Technical Information

NPAS POU 説明書（概要編）

TI 34P02P25-01

注意：

本 TI(Techinical Information)は，IM 34P02P25-01「NPAS POU 説明書（概要編）9 版」を TI 化したものです。

本文のフッタなどには，IM 番号がそのまま記載されていますが，TI 番号に読み替えてください。

なお，IM 34P02P26-01「NPAS POU 説明書（機能詳細編）」についても同様に TI 化していますので，こちらも併せて参照ください。

YOKOGAWA
横河電機株式会社

TI 34P02P25-01
2005.05.18　初版（YK）
2014.03.31　8 版（YK）

Blank Page

はじめに

■ 本書について

本書では，ロジックデザイナ上で，プロセス制御ループを構築するために使用するNPAS POUの概要について説明します。
本書を理解するにあたり，FCN/FCJの基本的な機能と，リソースコンフィギュレータ，ロジックデザイナの基本的な操作についての知識が必要です。
なお，NPAS POUの機能詳細については，NPAS POUのオンライン ヘルプを参照してください。

■ 本書の構成

本書の構成を以下に示します。

●1. 概要

NPAS POUの全体像とNPAS POU内部の機能構成要素を説明します。

●2.NPAS POUの制御ループの構成

NPAS POUで制御ループを構築する場合のデータの流れとデータの設定方法について説明します。

●3. プログラミング

ロジックデザイナ上でのNPAS POUの使用方法とプログラミング方法について説明します。

All Rights Reserved Copyright © 2004, Yokogawa Electric Corporation

安全に使用するための注意事項

■ 本書に対する注意

(1) 本書は，最終ユーザまでお届けいただきますようお願いいたします。
(2) 本製品の操作は，本書をよく読んで内容を理解したのちに行ってください。
(3) 本書は，本製品に含まれる機能詳細を説明するものであり，お客様の特定目的に適合することを保証するものではありません。
(4) 本書の一部または全部を，無断で転載，複製することは固くお断りします。
(5) 本書の内容については，将来予告なしに変更することがあります。
(6) 本書の内容について，もしご不審な点や誤り，記載もれなどお気付きのことがありましたら，弊社営業部，またはお買い求めの代理店までご連絡ください。

■ 本製品の保護・安全および改造に関する注意

(1) 本製品および本製品で制御するシステムの保護・安全のため，本製品を取り扱う際は，本書の安全に関する指示事項に従ってください。なお，これらの指示事項に反する扱いをされた場合，弊社は安全性を保証いたしかねます。

“取扱注意”を示しています。製品においては，人体および機器を保護するために，取扱説明書を参照する必要がある場所に付いています。また，取扱説明書においては，感電事故など，取扱者の生命や身体に危険が及ぶ恐れがある場合に，その危険を避けるための注意事項を記述してあります。

“保護接地端子”を示しています。機器を操作する前に必ずグランドと接続してください。

“機能用接地端子”を示しています。機器を操作する前に必ずグランドと接地してください。

電源スイッチ“オン”状態を示します。

電源スイッチ“スタンバイ”状態を示します。

電源スイッチ“オフ”状態を示します。

(2) 当該製品および当該製品で使用するシステムに対する保護・安全回路を設置する場合は，当該製品外部に別途用意するようお願いいたします。
(3) 当該製品の部品や消耗品を交換する場合は，必ず弊社の指定品を使用してください。
(4) 当該製品を改造することは固くお断りします。

■ 本製品の免責について

(1) 弊社は，保証条項に定める場合を除き，当該製品に関していかなる保証も行いません。
(2) 当該製品の使用により，お客様または第三者が損害を被った場合，あるいは弊社の予測できない当該製品の欠陥などのため，お客様または第三者が被った損害およびいかなる間接的損害に対しても，弊社は責任を負いかねますのでご了承ください。

■ 本ソフトウェアについて

(1) 弊社は，保証条項に定める場合を除き，当該ソフトウェアに関していかなる保証も行いません。
(2) 当該ソフトウェアは，特定された 1 台のコンピュータでご使用ください。別のコンピュータに対して，ご使用になる場合は別途ご購入ください。
(3) 当該ソフトウェアを，バックアップの目的以外でコピーすることは，固くお断りいたします。
(4) 当該ソフトウェアの収められている DVD-ROM（オリジナルメディア）は，大切に保管してください。
(5) 当該ソフトウェアの逆コンパイル，逆アセンブルなど（リバースエンジニアリング）を行うことは固くお断りします。
(6) 当該ソフトウェアは，弊社の事前の承認なしに，その全部または一部を譲渡，交換，転貨などによって第三者に使用させることは，固くお断りします。

表記上の約束

■ 取扱説明書中の記号

取扱説明書中の記号は，以下の内容を示します。

● 取扱説明書の中で共通に使用する記号

＜ ＞で囲った文字

オンラインマニュアルにおいて文字列をクリックすると関連する箇所へジャンプする機能が設定されている部分を示します。

例：

＜目次＞ ＜索引＞

入力文字列

以下の書体の文字列は，ユーザが実際の操作において入力する内容を示します。

例：

FIC100.SV=50.0

△記号

ユーザが入力する文字列の中の空白文字（スペース）を示します。

例：

.AL△PIC010△-SC

｛｝で囲った文字

ユーザが入力する文字列の中の省略可能な文字列を示します。

例：

.PR△TAG｛△.シート名｝

● キー操作やボタン操作の説明の中で使用する記号

［ ］で囲った文字

キー操作やボタン操作の説明の中で，キーボード上のキー，ウィンドウ上のボタン名，また，ウィンドウ上に表示される項目を示します。

例：

機能を切り換えるには［ESC］キーを押します。

● コマンド文やプログラム文などの書式説明の中で使用する記号

コマンド文やプログラム文などの中では，記号は以下の内容を示します。

＜ ＞で囲った文字

ユーザが一定の規則に沿って任意に指定できる文字列を示します。

例：

#define ＜識別子＞ ＜文字列＞

…記号

直前のコマンドや引数が繰り返し可能であることを示します。

例：

GetPropertyEx(PropName As String, List1 [, List2, ...] As Integer) As Variant

［ ］で囲った文字

省略可能な文字列を示します。

例：

SetCv(NewValue As Variant,[Transaction ID As Long])

| |で囲った文字

ユーザが複数候補から任意に選択できる文字列を示します。

例：

Move | X
XY
Y |

■ Windows 画面の表記について

● ［ ］で囲んであるもの

メニューバーのメニュー，コマンド，テキストボックス名，ボタンです。

例：メニューバーの［アイコン（E）］-［ファイル名を指定して実行（F）］を選択します。

（メニューバーの［アイコン（E）］を選択し，プルダウンメニューの［ファイル名を指定して実行（F）］を選択します。）

● 「 」で囲んであるもの

グループ，アイコン，ウィンドウ，ダイアログボックスの名称です。

例：「システムランチャ」グループの「マニュアルビューア」をダブルクリックします。

■ シンボルマークについて

本書中のシンボルマークは，以下の内容を示します。

取扱注意

「取扱注意」を示しています。
感電事故など，取扱者の生命や身体に危険が及ぶ恐れがある場合に，その危険を避けるための注意事項を記述してあります。

警　告

“警告”を示しています。
ソフトウェアやハードウェアを損傷したり，システムトラブルになる恐れがある場合に，注意すべきことがらを記述してあります。

重　要

「重要」を示しています。
操作や機能を知る上で，注意すべきことがらを記述してあります。

補足

説明を補足するためのことがらを記述してあります。

参照

参照すべき項目などを記述してあります。
参照先が緑色で表示されている場合は，参照先箇所を呼び出すことができます。
参照先が黒色で表示されている場合は，参照先箇所を呼び出すことはできません。

■ 図の表記について

本書に記載されている図では，説明の都合上，強調や簡略化，または一部を省略していることがあります。
本書の画面図は，機能理解や操作監視に支障を与えない範囲で，実際の画面表示と表示位置や文字（大／小文字など）が異なる場合があります。

著作権および商標

■ 著作権

DVD-ROM に含まれているプログラムおよびオンラインマニュアルなどの著作権は弊社に帰属します。
オンラインマニュアルについては，その内容を改ざんできないように PDF のセキュリティを設定しています。プリンタへの出力は可能です。
オンラインマニュアルをプリンタで出力してご使用になる場合は，本製品を利用するためだけにご使用ください。オンラインマニュアルをプリンタで出力したものを使う場合，最新版との不一致が起こらないようにご注意ください。ご使用時には，DVD-ROM の最新版と版数が一致していることをご確認ください。
オンラインマニュアルをコピーしたり，第三者に譲渡，販売，頒布（パソコン通信のネットワークを通じて通信により提供することを含みます）することを禁止します。また，無断でビデオテープその他に登録，録画することも禁止します。

■ 商標ならびにライセンスソフトウェアについて

- STARDOM は，横河電機株式会社の商標です。
- ASTMAC，eCUBE，astnex，YEWMAC，DARWIN，DAQSTATION は，横河電機株式会社の登録商標です。
- Microsoft，Windows，Excel，Visual Basic，ActiveX は，米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- Adobe および Acrobat は，アドビシステムズ社の商標で，特定の法域で登録されています。
- Ethernet（イーサネット）は，米国ゼロックス社の登録商標です。
- IBM，IBM-PC/AT は，IBM Corporation の登録商標です。
- その他，本文中に使われている会社名・商品名は，各社の商標または登録商標です。
- 本文中の各社の登録商標または商標には，TM，®マークは表示しておりません。

NPAS POU 説明書（概要編）

IM 34P02P25-01　9 版

目次

はじめに i
安全に使用するための注意事項 ii
表記上の約束 iv
著作権および商標 vii
1. 概要 1
1.1 NPAS POU 概要 2
1.1.1 NPAS POU 一覧 4
1.1.2 ライブラリー覧 8
1.2 NPAS POU 内部機能 9
1.3 NPAS POU で使用するデータ構造とグローバル変数 11
2. NPAS POU の制御ループの構成 13
2.1 制御データの流れ 14
2.2 読み返しデータの流れ 16
2.3 アクセスパラメータの参照・設定 18
2.4 エンジニアリングパラメータの設定 23
3. プログラミング 25
3.1 プロジェクトの作成 26
3.2 NPAS POU の設定 32
3.3 ロジックデザイナの基本的な操作 40
3.4 プロジェクトのコンパイル，ダウンロードとデバッグ 44
3.5 プログラミング上の留意点 58
3.6 オンラインダウンロード 65
付録 A グローバル変数一覧 69
付録 B デバイスラベルデータ 75
付録 C ロジックデザイナの FU と FB 77
付録 D NPAS POU と PAS POU との差分 85
取扱説明書　　改訂情報 i

Blank Page

1. 概要

NPAS POU とは，ロジックデザイナ上で使用するプロセス制御用の POU（Program Organization Unit：プログラム構成単位）です。
NPAS POU は，PID 制御などプロセス制御ループを作成するときに使用します。
本章では，NPAS POU の概要を説明します。

1.1 NPAS POU概要

本節では，NPAS POU 全体の概要を説明します。

プロセス制御で使用する NPAS POU の一般的な接続例と各々の POU の機能を以下に示します。

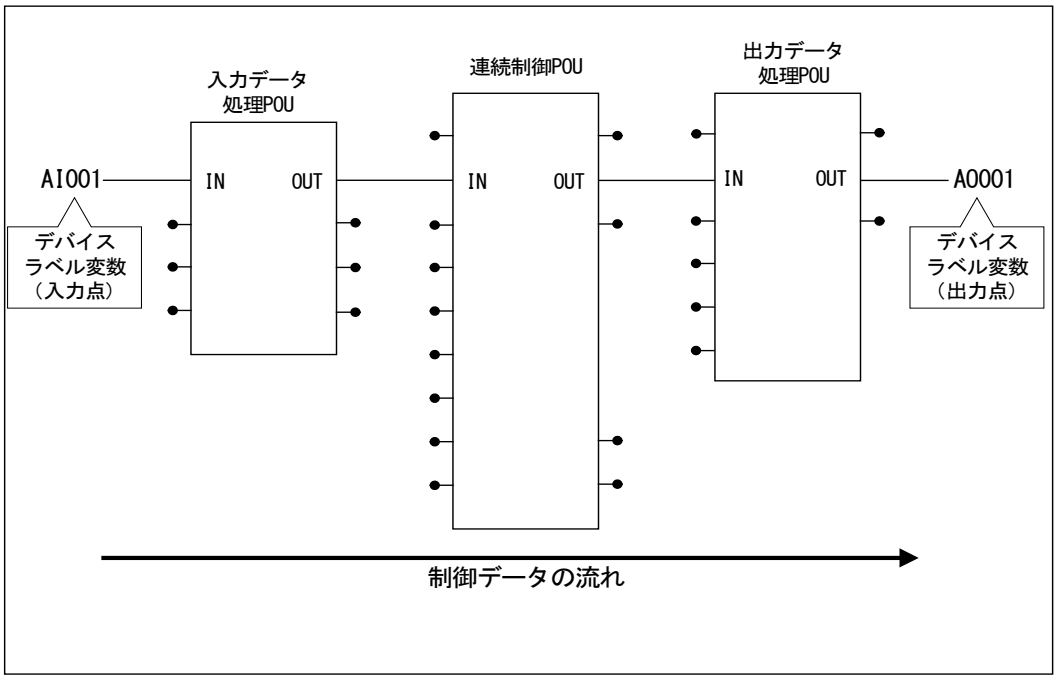


図 NPAS POU の接続例

● 入力データ処理 POU

入力データ処理 POU は，プロセスからの入力データを受けて，以下の処理などを行う POU です。

- プロセス入力データを工業量データに変換
- 入力オープンアラームチェック

● 連続制御 POU

連続制御 POU は，PID 制御演算などプロセス制御の中心となる演算を行う POU です。連続制御 POU では，以下の処理などを行います。

- 制御演算（PID 演算など）
- 入力処理（積算処理など）
- 出力処理（出力リミッタ，出力変化率リミッタなど）
- アラーム処理

● 出力データ処理 POU

出力データ処理 POU は，連続制御 POU からの出力信号を受けて，以下の処理などを行います。

・ 工業量データをプロセス出力データへ変換
・ タイトシャット，フルオープン処理

補足

デバイスラベル変数とは，ロジックデザイナ上で定義する変数で，プロセス入出力の入力点／出力点を示します。
デバイスラベル変数は，FCN/FCJ 上での制御アプリケーション実行時にリソースコンフィギュレータで設定したデバイスラベル（実 I/O）に結合されます。

■ その他の POU

NPAS POU では，上記の基本的なプロセス制御用 POU のほかに補足的な機能を行う POU も用意されています。

●数値演算 POU

数値演算 POU は，補正演算や移動平均演算などの数値演算処理を行う POU です。

●シーケンス制御 POU

シーケンス制御 POU は，スイッチ計器やタイマ，カウンタなどシーケンス処理に使用する POU です。

●FF-H1（フィールドバス）POU

FF-H1 POU は，FOUNDATION fieldbus 機器のパラメータの参照・設定のための POU です。

●ユーティリティ POU

ユーティリティ POU は，デジタルフィルタやデータ変換などの機能を持った POU です。

1.1.1 NPAS POU一覧

以下に NPAS POU 一覧を示します。

■ 入出力データ処理 POU

表　入出力データ処理 POU 一覧 (*1)

NPAS POU 名	説明
NPAS_AI_ANLG	標準アナログ入力処理 POU
NPAS_AI_DINT	DINT データアナログ入力処理 POU
NPAS_AI_FREQ	周波数入力処理 POU
NPAS_AI_TEMP	測定温度入力処理 POU
NPAS_AI_PULS_QT	定量積算パルス列入力処理 POU
NPAS_AI_PULS_CI	制御入力パルス列入力処理 POU
NPAS_AI_PCNT	正規化(%)データ入力処理 POU
NPAS_AI_REAL	REAL データ入力処理 POU
NPAS_AI_UDINT	UDINT データアナログ入力処理 POU
NPAS_AO_ANLG	標準アナログ出力処理 POU
NPAS_AO_DINT	DINT データアナログ出力処理 POU
NPAS_AO_PCNT	正規化(%)データ出力処理 POU
NPAS_AO_REAL	REAL データ出力処理 POU
NPAS_AO_UDINT	UDINT データアナログ出力処理 POU
NPAS_DI_STS	ステータス入力処理 POU
NPAS_DI_PUSHB	押しボタン入力処理 POU
NPAS_DI_WORD	WORD データ接点入力処理 POU
NPAS_DO_STS	ステータス出力処理 POU
NPAS_DO_STS_PW	パルス幅出力処理 POU（積算リセット型）
NPAS_DO_STS_PW2	パルス幅出力処理 POU（積算継続型)
NPAS_DO_STS_PWH	高分解能パルス幅出力処理 POU
NPAS_DO_STS_TP	時間比例オンオフ出力処理 POU
NPAS_DO_WORD	WORD データ接点出力処理 POU
NPAS_FFI_ANLG	FOUNDATION fieldbus H1(FF-H1)アナログ入力処理 POU
NPAS_FFI_STS	FF-H1 ステータス入力処理 POU
NPAS_FFO_ANLG	FF-H1 アナログ出力処理 POU
NPAS_FFO_STS	FF-H1 ステータス出力処理 POU
NPAS_AI_HART	HART バリアブル入力処理 POU

*1：　本 POU は PAS ポートフォリオライセンスなしで使用できます。

■ 連続制御 POU

表 連続制御 POU 一覧

NPAS POU 名	説明
NPAS_PVI	指示 POU
NPAS_PID	PID 調節 POU
NPAS_PI_HLD	サンプル PI 調節 POU
NPAS_ONOFF	2 位置式オンオフ調節 POU
NPAS_ONOFF_G	3 位置式オンオフ調節 POU
NPAS_MLD	手動操作 POU(出力プッシュバックなし)
NPAS_MLD_PB	出力値プッシュバック型 MLD
NPAS_MLD_BT	バイアストラッキング型 MLD
NPAS_RATIO	比率設定 POU
NPAS_RATIO_RT	比率設定 POU（レシオトラッキング機能付き）
NPAS_PG_L30	30 折れ線形プログラム設定 POU
NPAS_PG_L30_BP	30 折れ線形プログラム設定 POU（バンプレス機能付き）
NPAS_VELLIM	変化率制限 POU
NPAS_VELLIM_PB	出力値プッシュバック型 VELLIM
NPAS_AS_H/M/L	オートセレクタ POU
NPAS_FOUT	カスケード信号分配 POU
NPAS_FFSUM	フィードフォワード信号加算 POU
NPAS_FFSUM_BL	フィードフォワード信号加算 POU（バランス動作あり）
NPAS_XLMT_S	シングルクロス式リミット演算 POU
NPAS_XLMT_D	ダブルクロス式リミット演算 POU
NPAS_BSET_F	流量計量用定量設定 POU
NPAS_BSET_LW	質量計量用定量設定 POU

■ 数値演算 POU

表数値演算 POU 一覧

NPAS POU 名	説明
NPAS_LDLAG	進み遅れ POU
NPAS_DLAY	むだ時間 POU
NPAS_AVE_M	移動平均 POU
NPAS_AVE_C	区間平均 POU
NPAS_FUNC_VAR	可変折れ線関数 POU
NPAS_TP_CFL	温圧補正 POU
NPAS_T_CFL	温度補正 POU
NPAS_P_CFL	圧力補正 POU
NPAS_ASTM1	ASTM 補正 POU（旧 JIS）
NPAS_ASTM2	ASTM 補正 POU（新 JIS）
NPAS_SW13(SW31)	1 対 3(3 対 1)接点 1 回路切り替えスイッチ POU
NPAS_SW19(SW91)	1 対 9(9 対 1)接点 1 回路切り替えスイッチ POU
NPAS_BDBUF_R	数値データ格納 POU（REAL）
NPAS_BDBUF_T	時間データ格納 POU（TIME）

■ シーケンス制御 POU

表 シーケンス制御 POU 一覧

NPAS POU 名	説明
NPAS_SIO*	スイッチ計器(SI-1,SI-2,SO-1,SO-2,SIO-11,SIO-12,SIO-21,SIO-22)
NPAS_TM	タイマ POU
NPAS_CT	カウンタ POU

■ FF-H1（フィールドバス）POU

表 FF-H1 POU 一覧

NPAS POU 名	説明
NPAS_FFRD_ANLG	FF-H1 アナログデータ参照 POU
NPAS_FFRD_STS	FF-H1 ステータスデータ参照 POU
NPAS_FFRD_DINT	FF-H1 整数データ参照 POU
NPAS_FFWT_ANLG	FF-H1 アナログデータ設定 POU
NPAS_FFWT_STS	FF-H1 ステータスデータ設定 POU
NPAS_FFWT_DINT	FF-H1 整数データ設定 POU

■ ユーティリティ POU

表 ユーティリティ POU 一覧

NPAS POU 名	説明
NPAS_SQRT_LC	開平演算低入力カット付き POU
NPAS_CDR_DESTR	アナログ非構造体化 POU
NPAS_CDR_STR	アナログ構造体化 POU
NPAS_BPLS_SW	バンプレス切替 POU
NPAS_CDB_DESTR	デジタル非構造体化 POU
NPAS_CDB_STR	デジタル構造体化 POU
NPAS_CDD_DESTR	整数非構造体化 POU
NPAS_CDD_STR	整数構造体化 POU
NPAS_BCD_CI16	BCD コード入力 POU
NPAS_BCD_CO16	BCD コード出力 POU
NPAS_SIOCHGPLS	接点信号パルス形変換用 POU
NPAS_DGFLT	デジタルフィルタ POU
NPAS_DFILE	汎用データファイル POU
NPAS_ADDSW	INT 型パラメータ補助 POU
NPAS_ADDINT	
NPAS_SUBSW	
NPAS_SUBINT	
NPAS_AI2SW_A	二重化 AI 切替 TYPE A POU
NPAS_AI2SW_B	二重化 AI 切替 TYPE B POU
NPAS_AI3SW	三重化 AI 切替 POU
NPAS_SL2SW_A	大小レンジ入力切替 TYPE A POU
NPAS_SL2SW_B	大小レンジ入力切替 TYPE B POU
NPAS_DI3	三重化 DI 回路 POU
NPAS_NPS_STR	積算値構造体化 POU
NPAS_PB6	押しボタン POU
NPAS_RS8_A	8 入出力リソーススケジューラ(許可数非保持形)POU
NPAS_RS8_B	8 入出力リソーススケジューラ(許可数保持形)POU

1.1.2 ライブラリー覧

NPAS POU は，ロジックデザイナ用のライブラリの形で提供されます。
以下に NPAS POU のライブラリー覧を説明します。

NPAS POU のライブラリは，ロジックデザイナでのプロジェクト作成時に NPAS POU のテンプレートを選択することで自動的に登録されますので，通常は意識する必要はありません。
しかし，NPAS POU 以外のテンプレート選択した後に NPAS POU を使用する場合は，ライブラリを登録する必要があります。
以下に NPAS POU を提供するライブラリの一覧を示します。

■ インタフェースライブラリー覧

インタフェースライブラリは，NPAS POU の各 POU を提供するライブラリです。

表 インタフェースライブラリー覧

ライブラリ名	説明
SD_NPASPOU_PF	入出力データ処理 POU，連続制御 POU，数値演算 POU，シーケンス処理 POU を提供するライブラリです。
SD_NPASUTIL_PF	ユーティリティ POU を提供するライブラリです。
SD_NPASFFH1_PF	FF-H1 POU を提供するライブラリです。

■ 内部ライブラリー覧

内部ライブラリは，NPAS POU が内部的に使用している POU を提供するライブラリです。
内部ライブラリは，直接ユーザが使用するものではありませんが，上記，インタフェースライブラリ登録時は一緒に登録する必要があります。

表 内部ライブラリー覧

ライブラリ 名	説明
SD_PASPARTS_PF	SD_NPASPOU_PF の内部処理で使用されている POU の集まり
SD_FCXSYS_LIB SD_FCXSTD_LIB	NPAS POU で使用するためのコントローラに関連した情報の取得などを行う POU の集まり

補足

表の内部ライブラリは，NPAS POU と PAS POU で共通です。PAS POU のインタフェースライブラリは，SD_PASPOU_PF，SD_PASUTIL_PF，SD_PASFFH1_PF，SD_FCXPAS_LIB となります。

1.2 NPAS POU内部機能

本節では，個々の NPAS POU の内部機能についての概要を説明します。

NPAS POU は，以下の要素から構成されています。

- 入出力パラメータ
- 入力処理
- 制御演算処理
- 出力処理
- アラーム処理
- アクセスパラメータ
- エンジニアリングパラメータ

下図で連続制御 POU を例に NPAS POU の内部機能構成を示します。

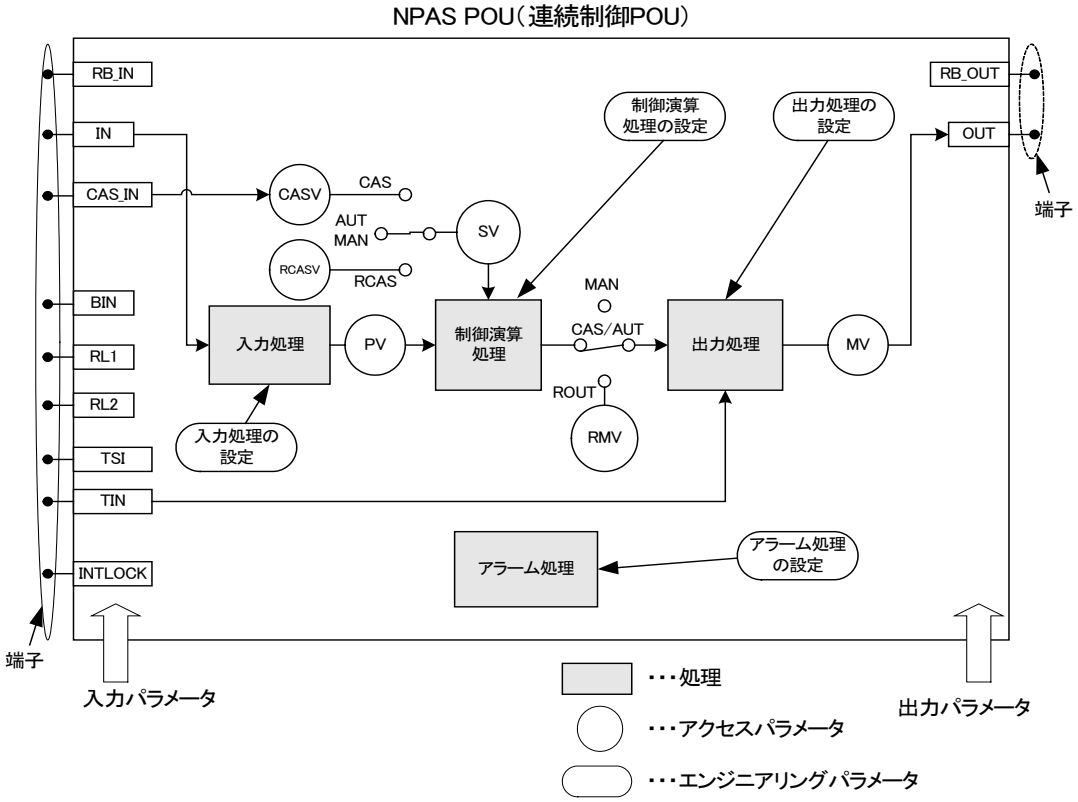


図 連続制御 POU の内部機能構成

● 入出力パラメータ

入出力パラメータは，外部との入出力を行うためのパラメータです。NPAS POU の外見上としては，入出力パラメータには端子があり，端子に他の POU や変数を結線することによりデータの入出力を行います。

● 入力処理

NPAS POU の入力処理とは，各入力パラメータから読み込まれたデータに対して，演算処理の前に実行する処理の総称です。NPAS POU の種類に応じて入力処理は異なります。入力処理の代表的なものとしては，以下のものがあります。

- キャリブレーション
- 積算
- 非定常時の入力処理

● 制御演算処理

PID 演算処理など制御演算処理の総称です。

● 出力処理

出力処理とは，制御演算により得られた値を出力パラメータに書き出す前に実行する処理の総称です。出力処理の代表的なものとしては，以下のものがあります。

- 出力リミッタ
- 出力変化率リミッタ
- 出力クランプ
- 出力値トラッキング
- 非定常状態の出力処理
- CPV プッシュバック

● アラーム処理

アラーム処理とは，測定値（PV）や操作出力値（MV）などの値からプロセスの異常を検出して，NPAS POU のアラーム状態に反映するとともに，検出の結果をまとめてメッセージとして操作監視機能に通知する機能です。アラーム処理は各 NPAS POU に存在します。

アラーム処理は，主に以下の 2 つの機能で構成されています。

- プロセスの異常を検出する「アラーム検出機能」
- 検出の結果を通知する「アラーム通知機能」

● アクセスパラメータ

アクセスパラメータは，PV，SV，MV など NPAS POU が動作するために必要な内部データの中で，外部からアクセス可能なデータです。これらのデータは，VDS/ASTMAC などの操作監視機能によりアクセスされます。

また，これらのデータはシーケンスなどからアクセスすることも可能です。

● エンジニアリングパラメータ

エンジニアリングパラメータは，NPAS POU のエンジニアリング時に決定する各種機能の設定パラメータです。

NPAS POU では，エンジニアリングパラメータにはデフォルト値が設定されています。そのため，エンジニアリング時には，すべてのデータを設定する必要はなく，デフォルト値から変更するパラメータのみを設定します。

1.3 NPAS POUで使用するデータ構造とグローバル変数

本節では，NPAS POU でエンジニアリングするときに使用する構造体データとグローバル変数について説明します。

■ 構造体データ

NPAS POU 間で受け渡されるデータは，単なるアナログ値などデータ値のみを持ったものではなく，データ値と同時にデータステータスやスケール上下限，工業単位などの複数のデータをまとめたデータを使用しています。
この複数のデータの固まりを構造体データと呼び，NPAS POU では，構造体データを受け渡すことにより，制御を行います。

以下に NPAS POU で使用する代表的な構造体データを示します。

● CData_REAL 型データ

CData_REAL 型データは，アナログ制御で使用するデータで，以下の構造を持ちます。

CData_REAL型構造体データ

Value	・・・データ値（REAL型，工業量データ）
Status	・・・データステータス(DWORD型)
CInfo	・・・結合情報(DWORD型，内部的に使用)
SH	・・・スケール上限値(REAL型)
SL	・・・スケール下限値(REAL型)
Unit	・・・工業単位(IndUnit型)

＊REAL型・・・32bit 浮動小数点データ
＊DWORD型・・・32bit ビット列データ
＊IndUnit型・・・8Byte 文字列データ

図 CData_REAL 型構造体データの構造

NPAS POU の CData_REAL 型データでは，数値（Value）は工業量データ（工業単位に基づく数値）を持ちます。
アナログ制御用 NPAS POU 間の入出力結合では，通常，この CData_REAL 型の構造体データが受け渡されます。

補足

上記の構造体データの中の各々のデータ(Value，Status，CInfo，SH，SL，Unit)を構造体データのメンバと呼びます。

● その他の代表的な構造体データ

NPAS POU 間の結合で使用するデータには，CData_REAL 型データの他に以下のものがあります。
以下に構造体データの名称とそのメンバを示します。

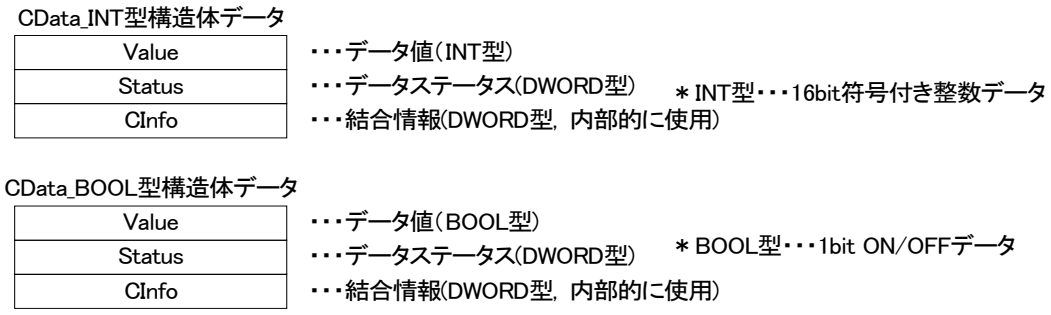


図 CData_INT 型，CData_BOOL 型の構造体データの構造

補足

上記の構造体データは，代表的な構造体データです。NPAS POU ではその他にも多くの構造体データが使用されています。その他の構造体データについては，その都度説明していきます。

■ システム用グローバル変数

グローバル変数とは，ワークシート内だけで使用するローカル変数に対して，FCN/FCJ 単位で共通に使用する変数です。グローバル変数は，ユーザが自由に作成できますが，グローバル変数の中には，システム用グローバル変数がデフォルトで定義されています。

NPAS POU を使用する場合，モードの設定やアラームステータスの判定などにシステム用グローバル変数を使用します。
以下に NPAS POU のシステム用グローバル変数について，代表的なものを紹介します。

表 システム用グローバル変数

グローバル変数	データ型	データ値	説明
GM_MODE_MAN	DWORD	00800000 (16 進数)	ブロックモード MAN
GM_MODE_AUT	DWORD	00400000 (16 進数)	ブロックモード AUT
GM_MODE_CAS	DWORD	00200000 (16 進数)	ブロックモード CAS
GM_ALRM_NR	DWORD	00800000 (16 進数)	アラーム状態 NR(Normal)
GM_ALRM_HH	DWORD	00080000 (16 進数)	アラーム状態 HH(High High Alarm)
GM_ALRM_LL	DWORD	00040000 (16 進数)	アラーム状態 LL(Low Low Alarm)
GM_ALRM_HI	DWORD	00008000 (16 進数)	アラーム状態 HI(High Alarm)
GM_ALRM_LO	DWORD	00004000 (16 進数)	アラーム状態 LO(Low Alarm)

参照

表のシステム用グローバル変数は代表的なもののみを示しています。システム用グローバル変数一覧は，付録 A に示しますので，必要に応じて参照してください。

2. NPAS POUの制御ループの構成

本章では，PID 制御ループを例に取り，NPAS POU での制御ループの構成を説明します。

■ PID 制御ループの構成例

以下に標準的な PID 制御ループを作成する場合の構成例を示します。

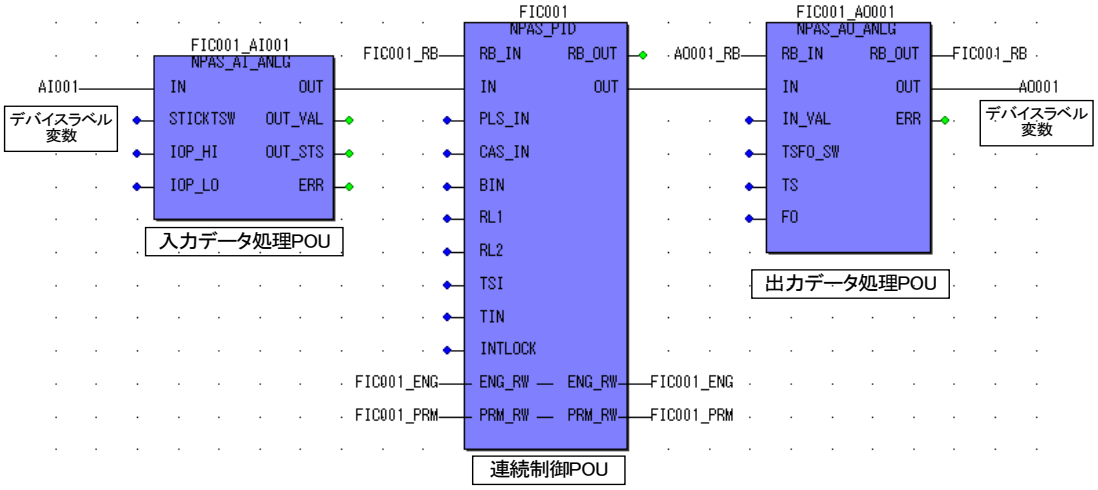


図 PID 制御ループの構成例

以降で，この PID 制御ループの構成例を元に各要素とそのデータの流れを説明します。

補足

ノイズ除去が必要な場合は，入力データ処理 POU の OUT 端子と連続制御 POU の IN 端子の間にデジタルフィルタ(NPAS_DGFLT)を配置します。デジタルフィルタはノイズを低減するための一次遅れフィルタです。図では，説明を簡略化するために省略しています。

2.1 制御データの流れ

本節では，PID 制御ループでの中心となる制御データの流れと，各データの構造体データを示します。

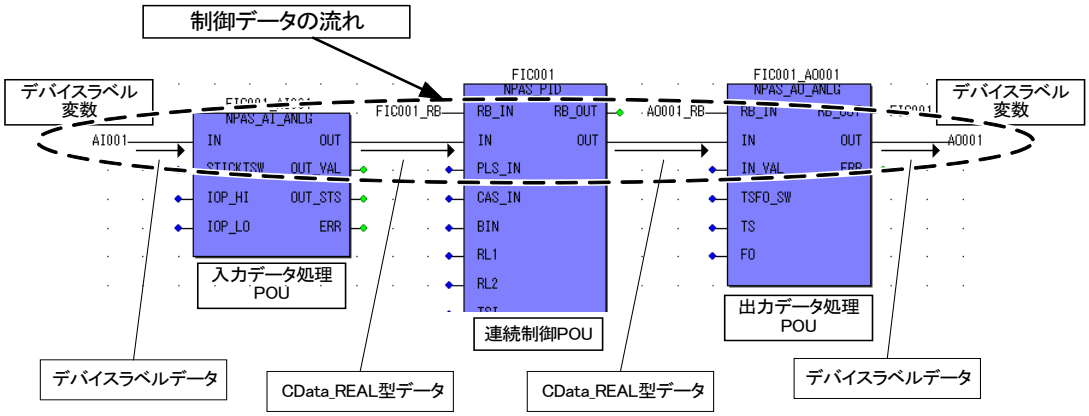


図 制御データの流れと構造体データ

制御データは，入力のデバイスラベル変数から入力され各 POU の IN 端子，OUT 端子を経由して，出力のデバイスラベル変数へ出力されます。

図のように制御データは，その流れの中でデータ型(構造)が変わります。
単に PID 制御ループを結線のみで構築する場合は，データ型を意識する必要はありませんが，ユーザ定義変数を経由して POU 間を結線する場合やシーケンスでデータ値を参照したい場合は，このデータ型を意識する必要があります。

以下に各データ型について説明します。

● CData_REAL 型データ

CData_REAL 型データは，工業量データとデータステータスやスケール上下限，工業単位を持つデータです。

補足

アクセスパラメータ PV，SV は，スケールと工業単位を持ちます。NPAS POU では，この PV，SV のスケールと工業単位は，IN 端子へ入力される CData_REAL 型データを参照して決定されます。

● デバイスラベルデータ

デバイスラベルデータとは，FCN/FCJ の実 I/O から入出力されるデータにスケール上下限や工業単位などを付加された構造体データです。
デバイスラベルデータは，標準アナログ入力(4-20mA/1-5V 入力)や測定温度入力(熱電対/測温抵抗体入力)など I/O の形式ごとに，構造体のデータの持ち方が異なります。

入出力データ処理 POU はデバイスラベルデータごとに用意され，CData_REAL 型データなどへの変換を行っています。

参照

デバイスラベルデータ一覧と，デバイスラベルごとに使用する入出力データ処理 POU は，付録 B に示しますので，必要に応じて参照してください。

2.2 読み返しデータの流れ

本節では，出力先の状態を取り込むための読み返しデータの流れを説明します。

連続制御では，出力値トラッキングやバンプレス処理など，出力先の状態により動作を切り替える必要があります。
IEC61131-3 では，信号は必ず左から右へ流れると規定されているので，出力先の状態を参照するために，制御データとは別に読み返しデータを設定する必要があります。

■ 操作出力の読み返し

出力先の状態の読み返しは，連続制御 POU などの操作出力に関して行います。操作出力である連続制御 POU－出力データ処理 POU－デバイスラベル（出力）間の読み返しデータの流れを下図に示します。

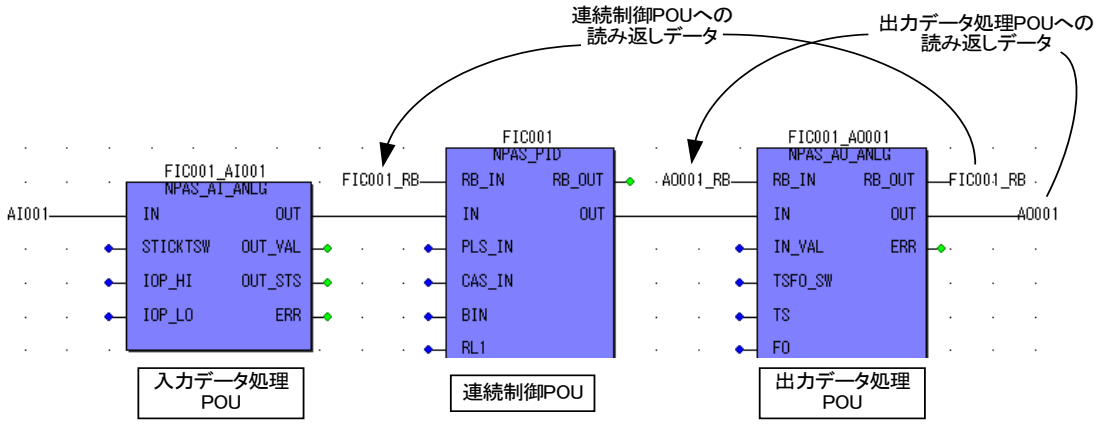


図 読み返しデータの流れ

連続制御 POU への読み返しデータとして，出力データ処理 POU の RB_OUT 端子の出力を連続制御 POU の RB_IN 端子へ，ユーザ定義変数 FIC001_RB を経由して接続しています。
また，出力データ処理 POU の読み返しデータとして，デバイスラベル変数の読み返し変数 AO001_RB を出力データ処理 POU の RB_IN 端子へ接続しています。

補足

デバイスラベル変数用の読み返し変数 AO001_RB は，デバイスラベル変数 AO001 を定義したときに自動的に生成されます。

重 要

読み返しデータの設定は，NPAS POU のエンジニアリング上，非常に重要です。読み返しデータを設定しなくてもコンパイル時にエラーは出ませんが，正常に制御ができません。そのため，必ず読み返しデータを設定してください。

● 読み返しデータのデータ型

読み返しデータも構造体データとなります。
読み返しデータのデータ型については，連続制御 POU への読み返しデータ（図の FIC001_RB のデータ）は CData_REAL 型，出力データ処理 POU への読み返しデータ（図の AO001_RB）はデバイスラベルデータとなります。

補足

アクセスパラメータ MV はスケールと工業単位を持ちます。NPAS POU では，この MV のスケールと工業単位は，RB_IN 端子へ入力される読み返しデータ（CData_REAL 型データ）を参照して決定されます。

● カスケード接続時の読み返しデータ

カスケード接続時も下位の連続制御 POU と上位の連続制御 POU 間で読み返しデータを結合する必要があります。以下にカスケード接続時の読み返しデータの接続例を示します。

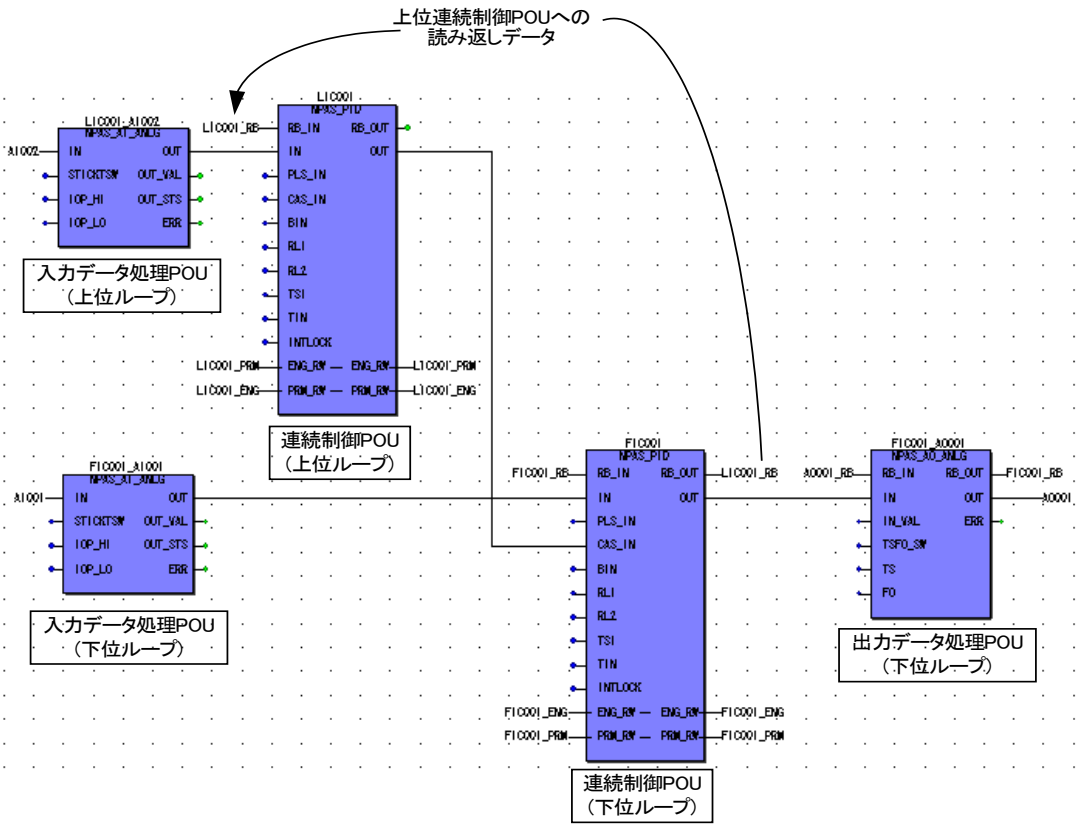


図 カスケード接続時の読み返しデータ

図のようにカスケード接続時には，下位の連続制御 POU の RB_OUT 端子の出力を上位の連続制御 POU の RB_IN 端子へ，ユーザ定義変数 LIC001_RB を経由して接続します。

2.3 アクセスパラメータの参照・設定

本節では，アクセスパラメータの参照・設定の方法について説明します。

NPAS POU では，内部的に PV，SV，MV などの値や MAN/AUT/CAS などのブロックモード，アラーム状態などのデータを持ちます。
NPAS POU では，これらの NPAS POU で内部的に保持しているデータをアクセスパラメータと呼びます。

通常は，アクセスパラメータはオペレーション時に参照・設定するもので，VDS/ASTMAC などの操作監視機能よりアクセスされるものですが，シーケンスなどからこのアクセスパラメータを参照・設定することも可能です。
シーケンスなどから参照・設定する場合には，アクセスパラメータ参照・設定用変数を使用して，各アクセスパラメータの参照や設定を行います。

このアクセスパラメータ参照・設定用変数は，以下の図の"FIC001_PRM"ようにユーザ定義変数を定義して使用します。

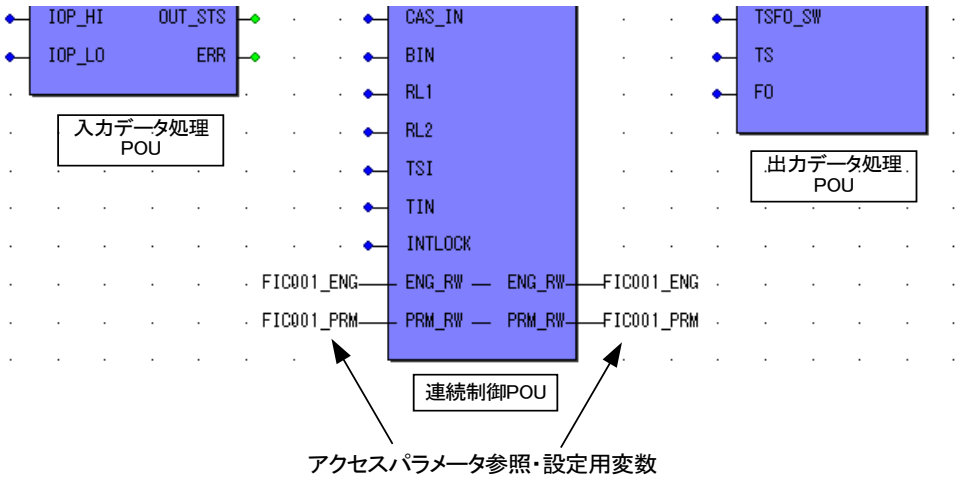


図 アクセスパラメータ参照・設定用変数

補足

アクセスパラメータ参照・設定用変数は，シーケンスなどで使用しない場合も必ず定義しておく必要があります。定義していない場合は，コンパイル時にエラーとなります。

■ アクセスパラメータ参照・設定用変数の構造

このアクセスパラメータ参照・設定用変数は構造体データとなっています。
その構造体データの主要なメンバを以下に示します。

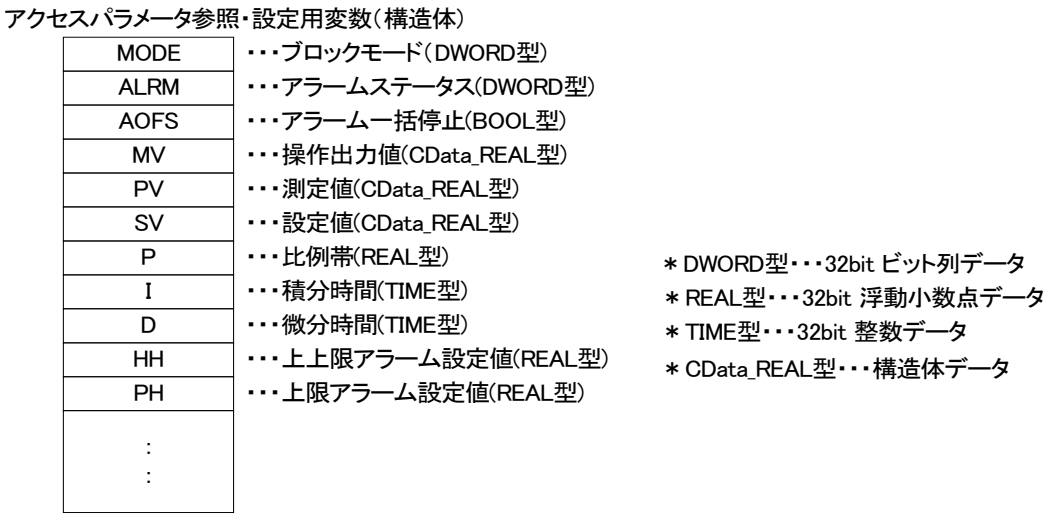


図 アクセスパラメータ参照・設定変数の構造

このようにアクセスパラメータ参照・設定用変数は，様々なメンバを持っています。
また，その中でも PV，SV，MV メンバは，さらに CData_REAL 型の構造体データとなり，Value や Status などのメンバを持っています。

参照

上記のアクセスパラメータ参照・設定用変数のメンバは代表的な例です。アクセスパラメータ参照・設定用変数のメンバは，NPAS POU の POU ごとに異なります。詳細は，各 POU の詳細説明を参照ください。

以降で，アクセスパラメータの参照例と設定例を示します。

■ アクセスパラメータ参照例

アクセスパラメータの参照では，アクセスパラメータ参照・設定用変数の変数名の後にドット(.)と構造体データのメンバ名を指定して参照します。
たとえば，変数名"FIC001_PRM"のブロックモードを参照する場合は，"FIC001_PRM.MODE"を指定します。
また，変数名"FIC001_PRM"の PV 値を参照する場合は，"PID_PRM.PV"と指定します。

以下にアクセスパラメータの参照例を示します。

● MODE の参照

連続制御 POU "FIC001"のブロックモードを参照し，MAN 時に TRUE(ON)を出力するシーケンスロジックは，以下のように設定します。

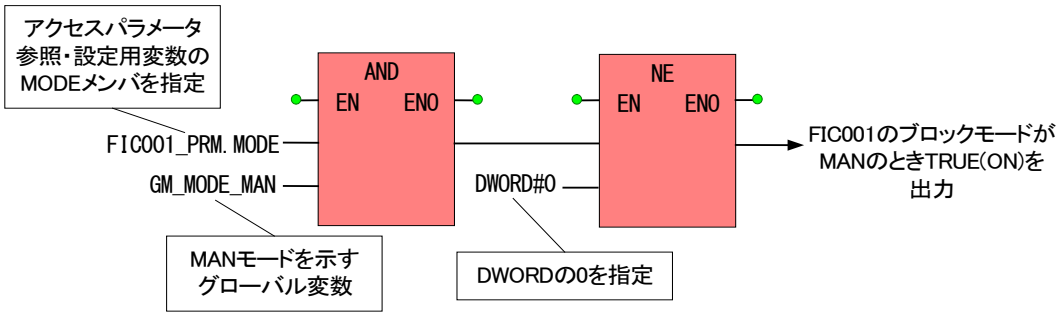


図 ブロックモード参照

図中の GM_MODE_MAN は，MAN モードを示す DWORD 値を持ったグローバル変数です。PAS POU のモードやアラーム状態は，DWORD 値のビット情報によりモードを示します。そのため，モードやアラーム状態を検出するときは，グローバル変数との AND をとり，その結果が，DWORD 値の 0 と NE（Not Equal）であるかどうかにより検出します。

補足

図中の AND ファンクションと NE（Not Equal）ファンクションは，ロジックデザイナで標準に用意されているファンクションで，2 つの入力端子の値を演算し，演算結果を出力端子から出力します。ロジックデザイナで用意されているファンクションやファンクションブロック一覧については，本 IM の付録 C に示しますので，参照してください。

● PV 値の参照

連続制御 POU "FIC001"の PV の値を他の POU で参照する場合は，以下のように設定します。

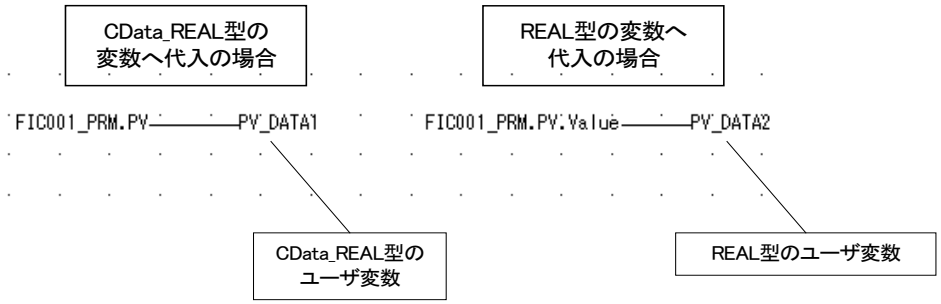


図 PV 値の参照

このようにスケール上下限，工業単位，データステータスを含めた CData_REAL 型のデータを参照する場合は，"FIC001_PRM.PV"を指定します。一方，REAL 型であるデータ値のみを参照したい場合は，"FIC001_PRM.PV.Value"を指定します。

■ アクセスパラメータ設定例

以下にアクセスパラメータの設定例を示します。
アクセスパラメータの設定もアクセスパラメータ参照・設定用変数の変数名の後にドット(.)と構造体データのメンバ名を指定します。

● ブロックモード(MODE)の設定

FIC001 のモードを AUT に設定する場合は，以下のように MOVE ファンクションを使用します。

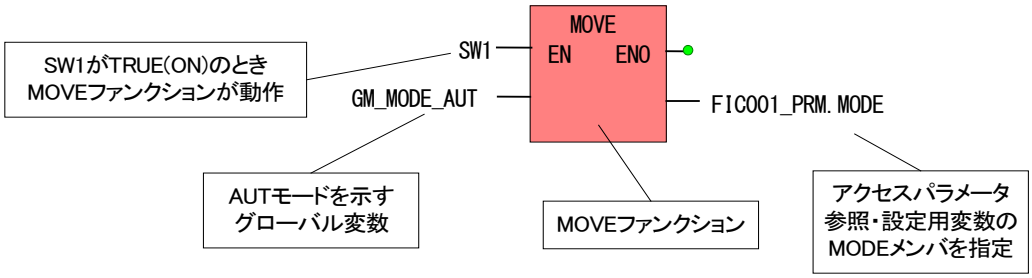


図 ブロックモードの設定

MOVE ファンクションは，EN 端子の入力が TRUE（ON）のとき入力端子からの入力値を出力端子の接続先へ出力します。ブロックモードの設定も AUT モードを示すグローバル変数（GM_MODE_AUT）を使用して設定します。

補足

MOVE ファンクションの EN 端子へ何も結線しない状態で使用すると，EN 端子への入力が常に TRUE（ON）であるとみなされ，常にブロックモードの設定が動作します。条件により MOVE ファンクションを動作させたい場合は，条件により TRUE／FALSE が切り替わる変数を EN 端子へ接続してください。

● MV 値の設定

FIC001 の MV 値を設定する場合も，以下のように MOVE ファンクションを使用します。

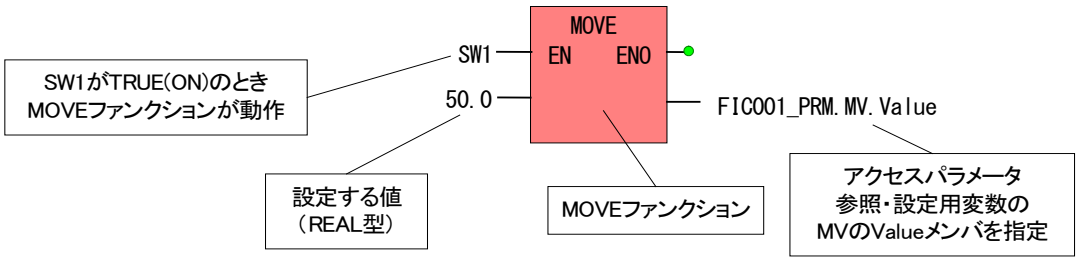


図 MV 値の設定

FIC001_PRM のメンバ MV は，CData_REAL 型の構造体を持っています。
データの設定は，その CData_REAL 型データの Value メンバに設定するので，"FIC001_PRM.MV.Value"を指定します。

補足

アクセスパラメータの CData_REAL 型，または，CData_INT 型，CData_BOOL 型データについては，データ値（Value メンバ）のみ設定が可能です。工業単位やデータステータスなど他のメンバの設定をシーケンスなどから行うことはできません。

2.4 エンジニアリングパラメータの設定

本節では，エンジニアリングパラメータの設定方法について説明します。

エンジニアリングパラメータとは，コメントやアラームヒステリシス，出力動作方向などのエンジニアリング時に必要な設定が格納されているパラメータです。

エンジニアリングパラメータの各データは，デフォルト値を持っています。
このエンジニアリングパラメータのデフォルト値から変更したい場合や，デフォルト値を持っていないコメントを設定する場合は，エンジニアリングパラメータ設定用変数を使用して設定します。

このエンジニアリングパラメータ設定用変数は，以下の図の"FIC001_ENG"ようにユーザ作成変数を定義して使用します。

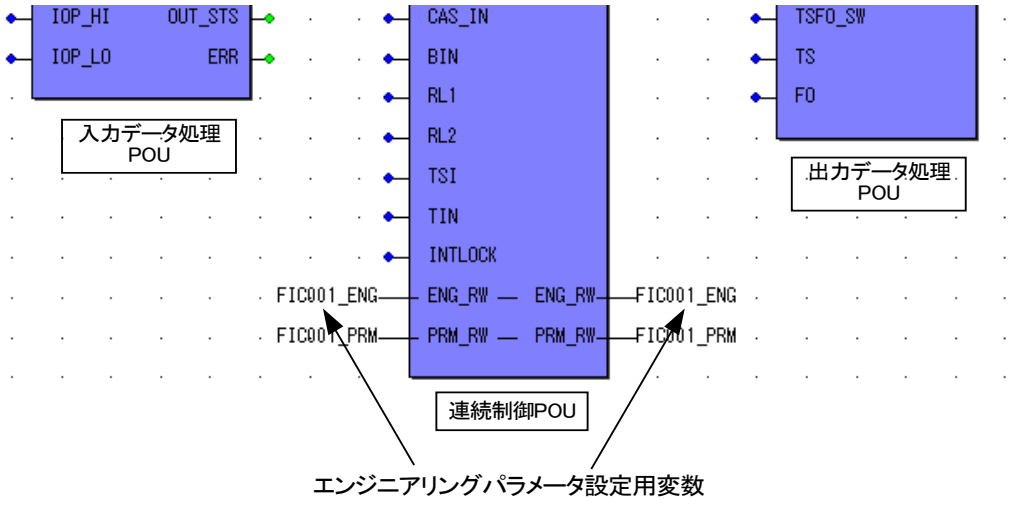


図 エンジニアリングパラメータ設定用変数

補足

エンジニアリングパラメータ設定用変数は，デフォルト値の変更を行わない場合も必ず定義しておく必要があります。定義していない場合は，コンパイル時にエラーとなります。

■ エンジニアリングパラメータ設定用変数の構造

エンジニアリングパラメータ設定用変数は構造体データとなっています。
その構造体データの主要なメンバを以下に示します。

表 代表的なエンジニアリングパラメータ

メンバ	データ型	説明	デフォルト値	設定範囲
COMMENT	STRING(32)	コメント	－	－
_CTL_DIR_SW	BOOL	制御動作方向正逆動作スイッチ	0	0（逆動作） 1（正動作）
_HYS_ALRM_HILO	REAL	上限/下限ヒステリシス	2.0	0.0～100.0%
_HYS_ALRM_HHLL	REAL	上上限/下下限ヒステリシス	2.0	0.0～100.0%
_HYS_ALRM_VEL	REAL	入力変化率ヒステリシス	2.0	0.0～100.0%
_DV_CHK_GAIN	REAL	偏差チェックフィルタゲイン	0.0	0.000～10.000
_DV_CHK_INTVL	TIME	偏差チェックフィルタ時定数	0 秒	0～10000 秒
_HYS_ALRM_DV	REAL	偏差アラームチェックヒステリシス	1.0	0.0～100.0%
_HYS_ALRM_MHML	REAL	出力上限/下限アラームヒステリシス	2.0	0.0～100.0%
_MV_VEL	REAL	変化率リミット値	100.0	0.0～100.0%
_SUM_SW	INT	積算指定スイッチ	0	0（積算なし） 1（秒積算） 2（分積算） 3（時積算） 4（日積算）

参照

上記エンジニアリングパラメータ設定用変数のメンバは，代表的な例です。エンジニアリングパラメータ設定用変数のメンバは，NPAS POU の POU ごとに異なります。詳細は，各 POU の詳細説明を参照ください。

■ エンジニアリングパラメータの設定例

エンジニアリングパラメータの設定例を示します。
エンジニアリングパラメータの設定では，エンジニアリングパラメータ設定用変数にメンバを指定して，設定値を代入します。

'流量調節' —— FIC001_ENG.COMMENT
TRUE —— FIC001_ENG._CTL_DIR_SW
INT#1 —— FIC001_ENG._SUM_SW
2.0 —— FIC001_ENG._MV_VEL

図 エンジニアリングパラメータの設定例

3. プログラミング

本章では，ロジックデザイナ上で NPAS POU を使用する際のプログラミング方法と注意点を説明します。

3.1 プロジェクトの作成

プロジェクトとは，ロジックデザイナ上でのエンジニアリングの単位を示します。ロジックデザイナ上で，FCN/FCJ のエンジニアリングを行う場合は，このプロジェクトを作成する必要があります。
本節では，プロジェクトの作成方法について説明します。

■ テンプレートによるプロジェクトの新規作成

NPAS POU を使用するプロジェクトを新規に作成する場合は，用意されたテンプレートプロジェクトを利用できます。
テンプレートを使ったプロジェクトの新規作成手順について説明します。

1. ロジックデザイナを起動します。
2. 「ファイル」－「新規プロジェクト」を選択すると，ダイアログが表示されます。

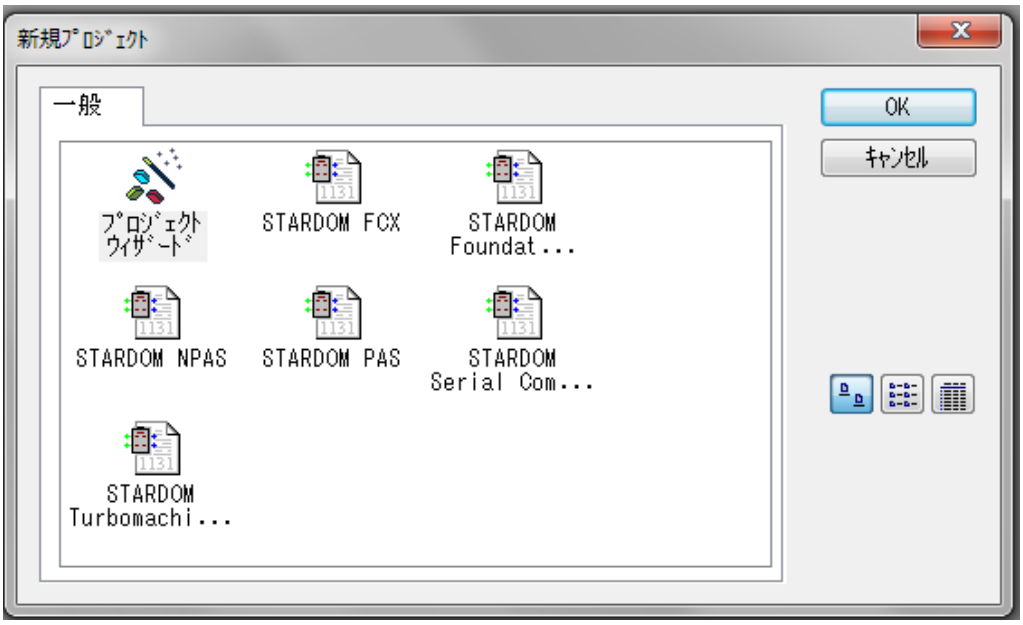


図 新規プロジェクト作成

新規プロジェクトを作成するための各種テンプレートが表示されます。

3.「STARDOM NPAS」を選択し，OK を押します。
NPAS POU を使用してアプリケーションを作成していくための外枠ができます。

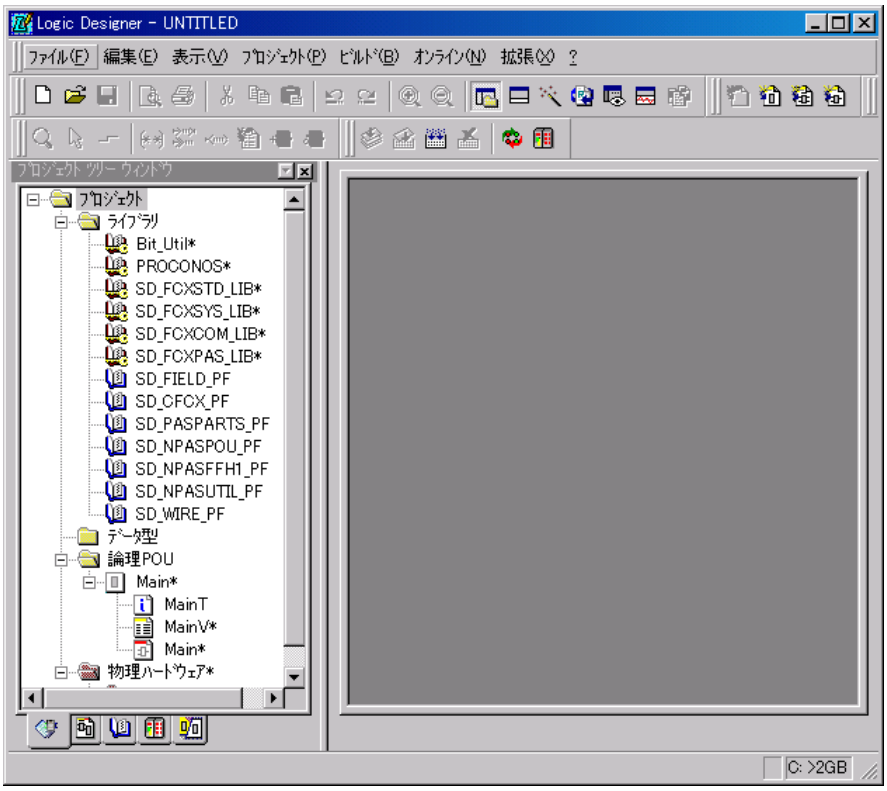
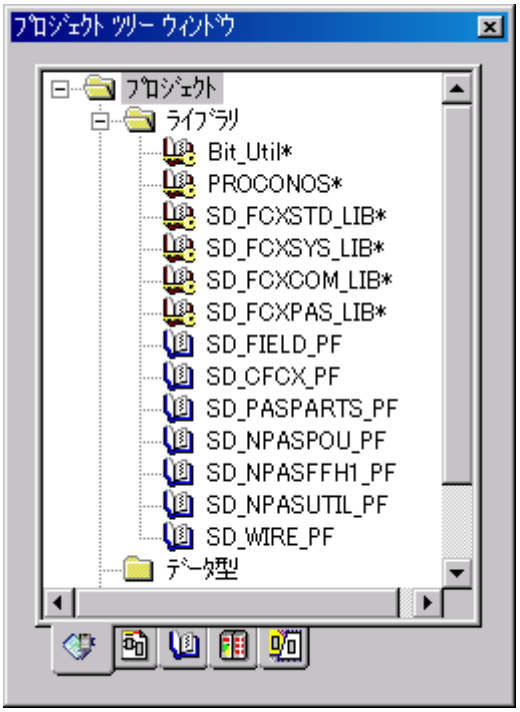


図 NPAS POU のテンプレートで作成したプロジェクト

［テンプレートプロジェクトの説明］

・ NPAS POU を使用するために必要なライブラリがインポートされています。



図ライブラリー覧

- "Main"という名前の 1 つのプログラム POU が定義され，タスクに割りつけてあります。
- 物理ハードウェアには，1 つのコンフィグレーション（Configuration），リソース（FCX01），タスク（Tasks）が作成されています。

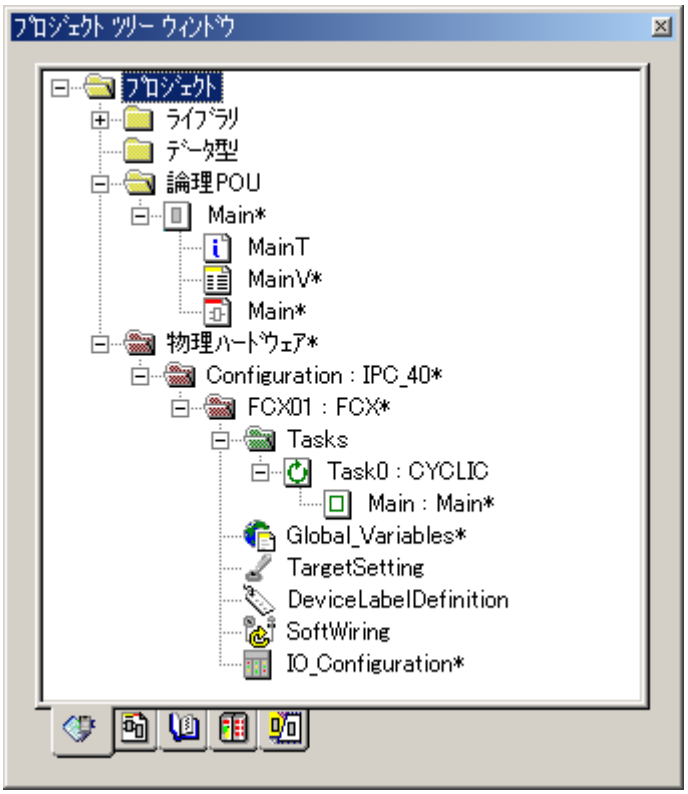


図 コンフィギュレーションにある各機能の一覧

■ ライブラリのインポート

テンプレートプロジェクトを使用すると，NPAS POU でアプリケーションを作成するために必要なライブラリがすでにインポートされています。
この他のライブラリを使用したい場合には，ユーザがライブラリのインポート操作を行います。その手順について説明します。

補足

NPAS POU のみを使用してプログラミングを行う場合は，ライブラリのインポートを行う必要はありません。他のアプリケーションポートフォリオのライブラリを使用する場合にインポートを行います。必要なライブラリの詳細については，それぞれのアプリケーションポートフォリオのオンラインヘルプを参照してください。

1. ライブラリをインポートしたいプロジェクトを開きます。
2. プロジェクトツリーの「ライブラリ」を右クリックし，「挿入」－「ライブラリ」を指定します。
3. ライブラリ選択ダイアログが表示されます。

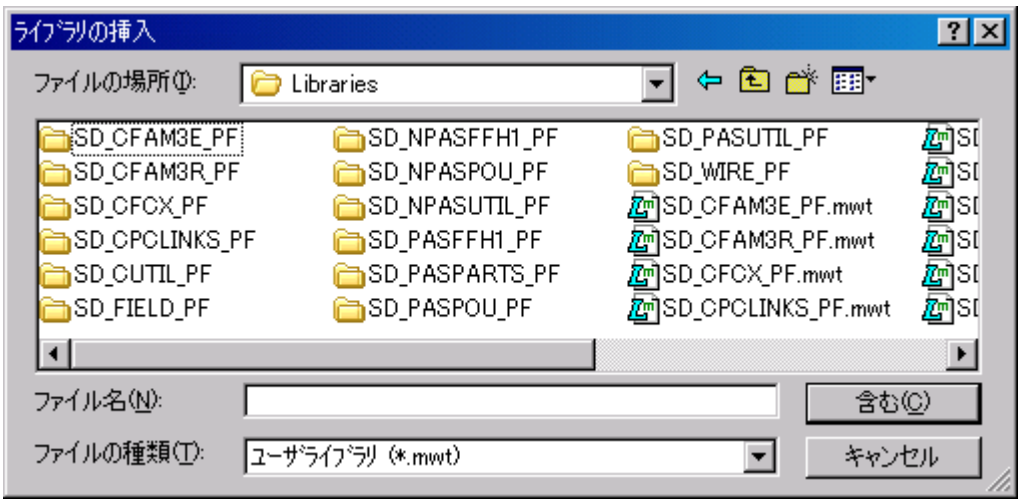


図 ライブラリ選択ダイアログ

ここで表示されるフォルダは，メニューバーの「拡張」－「オプション」の「ディレクトリ」タブにある「ライブラリのディレクトリ」に設定されているパスのものです。

4. インポートしたいライブラリを選択します。このとき，ライブラリ名のフォルダではなく， "mwt"という拡張子のファイルを選択してください。

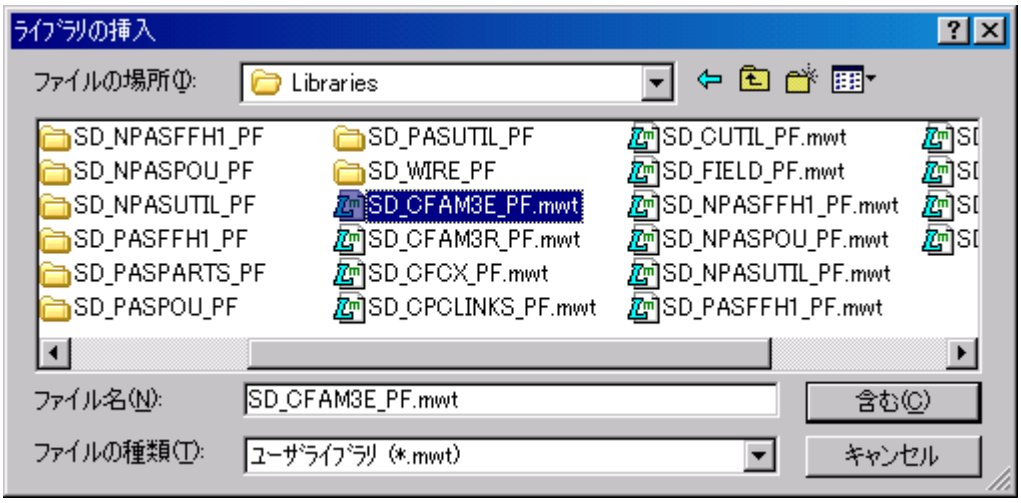


図 ライブラリ選択

5. 指定したライブラリが，プロジェクトツリーの「ライブラリ」の最後尾に追加されます。
6. ライブラリを追加した後は，プロジェクト全体に変更があったと認識されるので， メニューバーの「ビルド」－「メイク」を実行します。
7. ライブラリがインポートされました。

エディットウィザードを表示すると新たにインポートしたライブラリがグループ欄に表示されるようになります。
そのライブラリを選択するとそのなかに作成されている POU が一覧で表示されます。

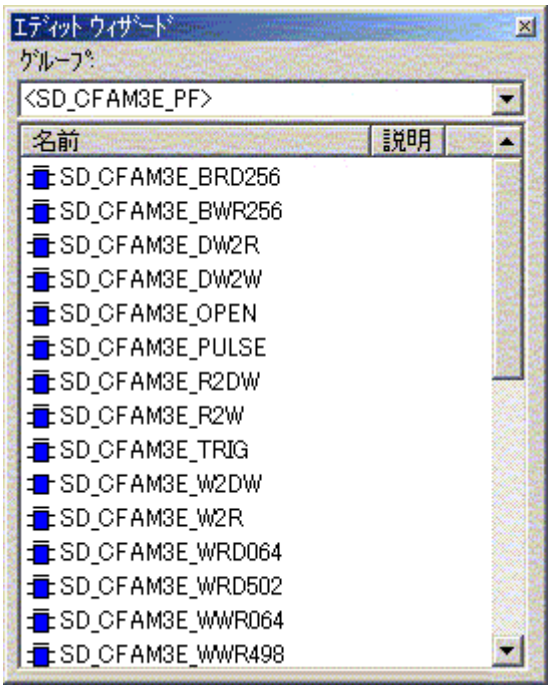


図 エディットウィザードの POU 一覧

■ グローバル変数

グローバル変数は，リソースの中の Global Variables をダブルクリックすると変数シートが展開し，内容を参照することができます。
システム用グローバル変数は，ほとんど Read Only の定数として使用しますので，安易に値を書き換えると，コントローラや NPAS POU が正常に動作しなくなることがあります。
中にはユーザで初期値を変更可能なものや，コントローラの実行中に値を変更して使用できるものがあります。

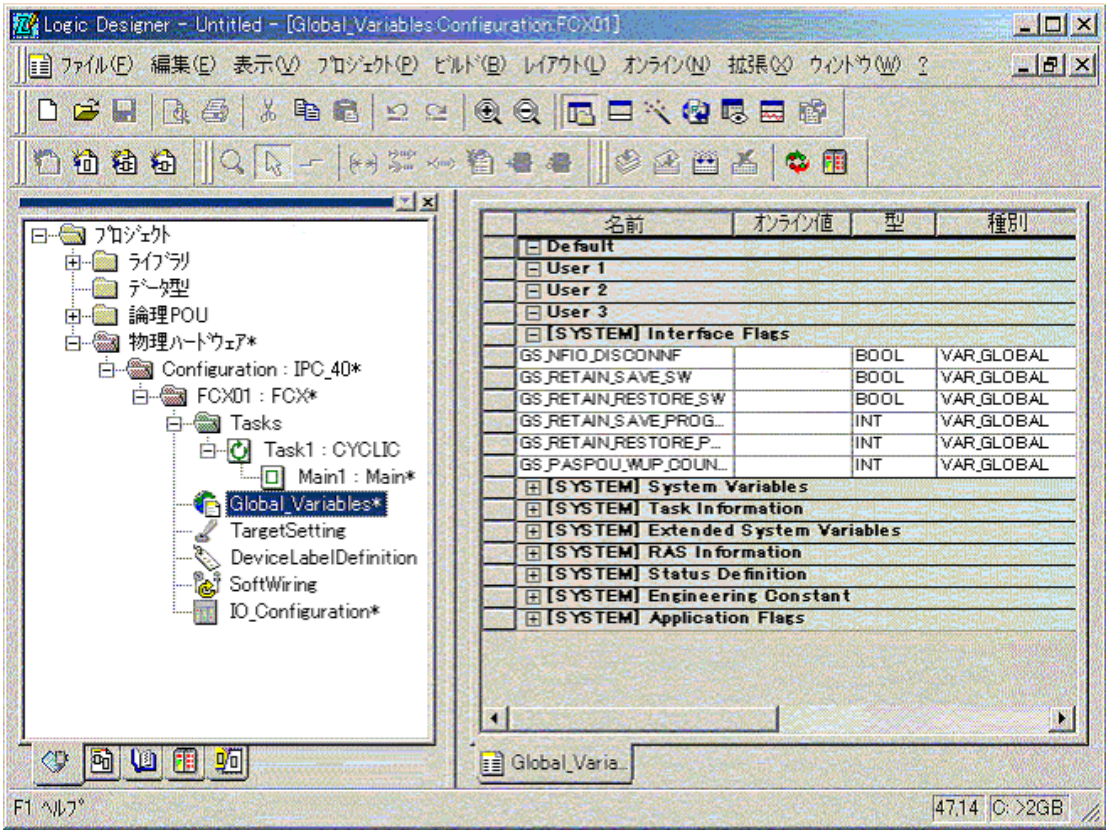


図 グローバル変数

3.2 NPAS POUの設定

NPAS POU では，"エンジニアリングパラメータ"と"アクセスパラメータ"の 2 種類のパラメータを持っています。
本節では，これらのパラメータに値を設定・参照する方法について説明します。

■ タグの作成

NPAS_PID のインスタンス（タグ）を作成する手順を説明します。

1. NPAS_PID を定義するプログラム POU のコードワークシート（例，Main）を開きます。
2. エディットウィザードで，SD_NPASPOU_PF グループを選択し，その中から NPAS_PID を選択します。

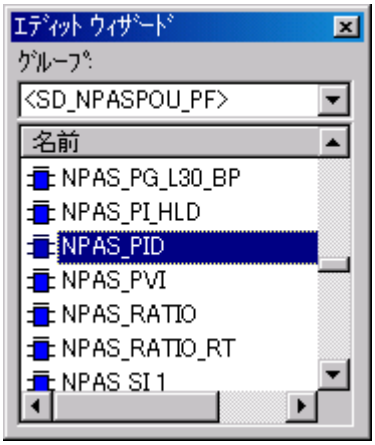


図 NPAS POU の選択

3. 変数のプロパティで，名前にタグ名（例，FIC001）を入力します。

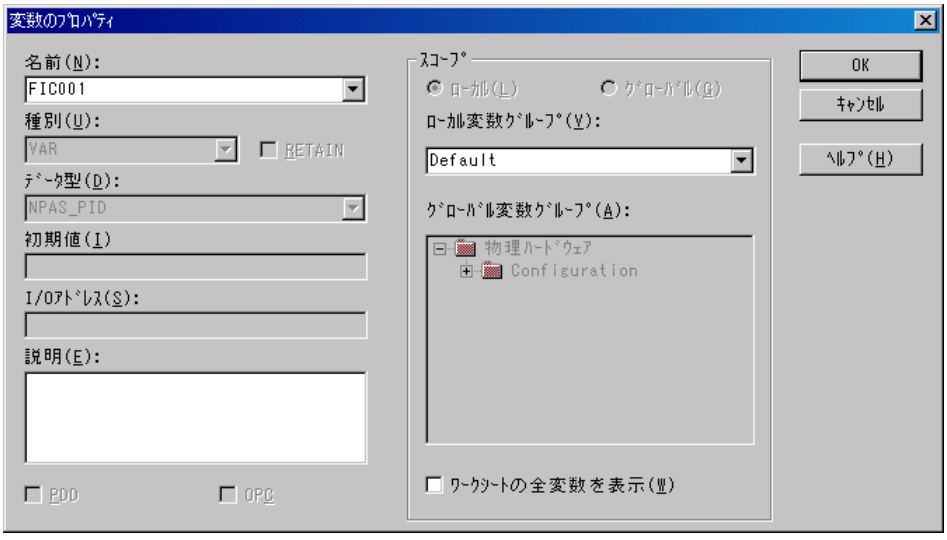


図 NPAS POU の選択

4. NPAS POU のインスタンス（タグ）の入出力パラメータ「ENG_RW」にエンジニアリングパラメータ設定用の変数（例，変数名 FIC001_ENG）を作成します。
5. NPAS POU のインスタンス（タグ）の入出力パラメータ「PRM_RW」にアクセスパラメータ参照・設定用の変数（例，変数名 FIC001_PRM）を作成します。

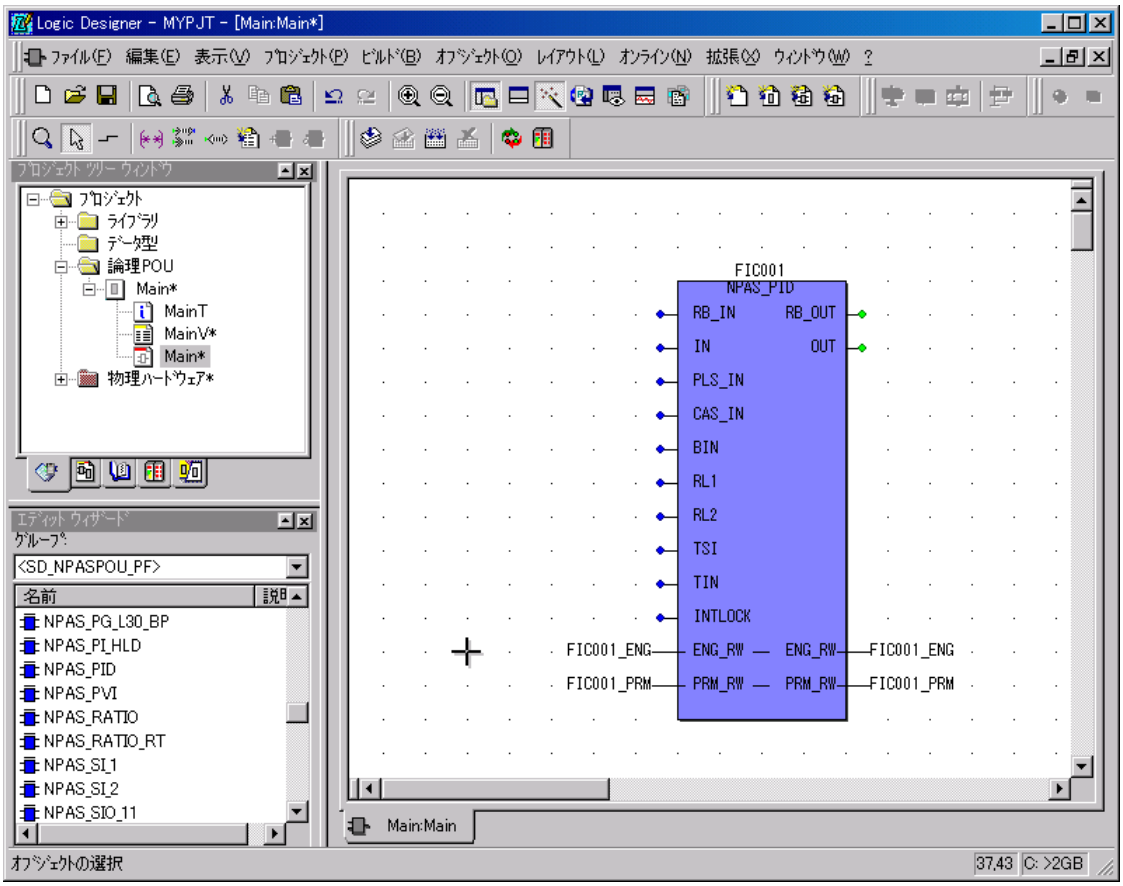


図 NPAS POU の作成

これで，NPAS_PID のインスタンス（タグ）が作成されました。

■ エンジニアリングパラメータの設定

NPAS_PID にエンジニアリングパラメータを設定する手順を説明します。
先に作成したエンジニアリング設定用の変数は，構造体になっており，NPAS POU に存在するすべてのエンジニアリングパラメータをメンバとして持っています。これらのデータを設定するときには，構造体メンバにアクセスする方法で値を設定します。
なお，エンジニアリングパラメータには，初期値から変更するパラメータのみ設定します。初期値のまま使用するパラメータの設定は不要です。

1. エンジニアリングパラメータ設定用の変数を指定します。
   変数プロパティの名前に，先に作成した変数名（例，変数名 FIC001_ENG）を入力し，更に，"."（ドット）を入力すると，エンジニアリングパラメータの一覧が表示されます。この一覧から設定するパラメータを選択します（例，コメントを設定する場合，COMMENT を選択）。

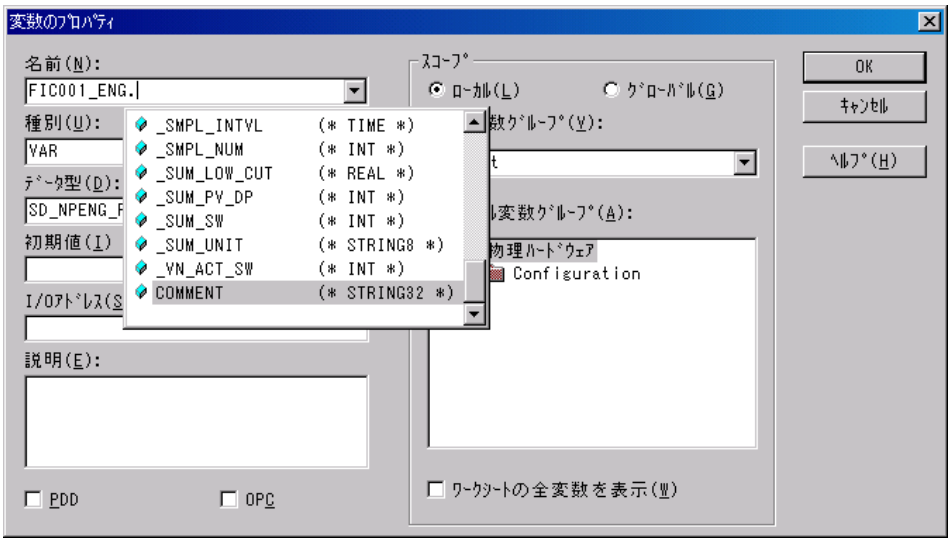


図 エンジニアリングパラメータの選択

2. エンジニアリングパラメータの設定値となる変数を作成し（例，コメントの設定値 '流量調節計'），先程指定した変数と結線します。

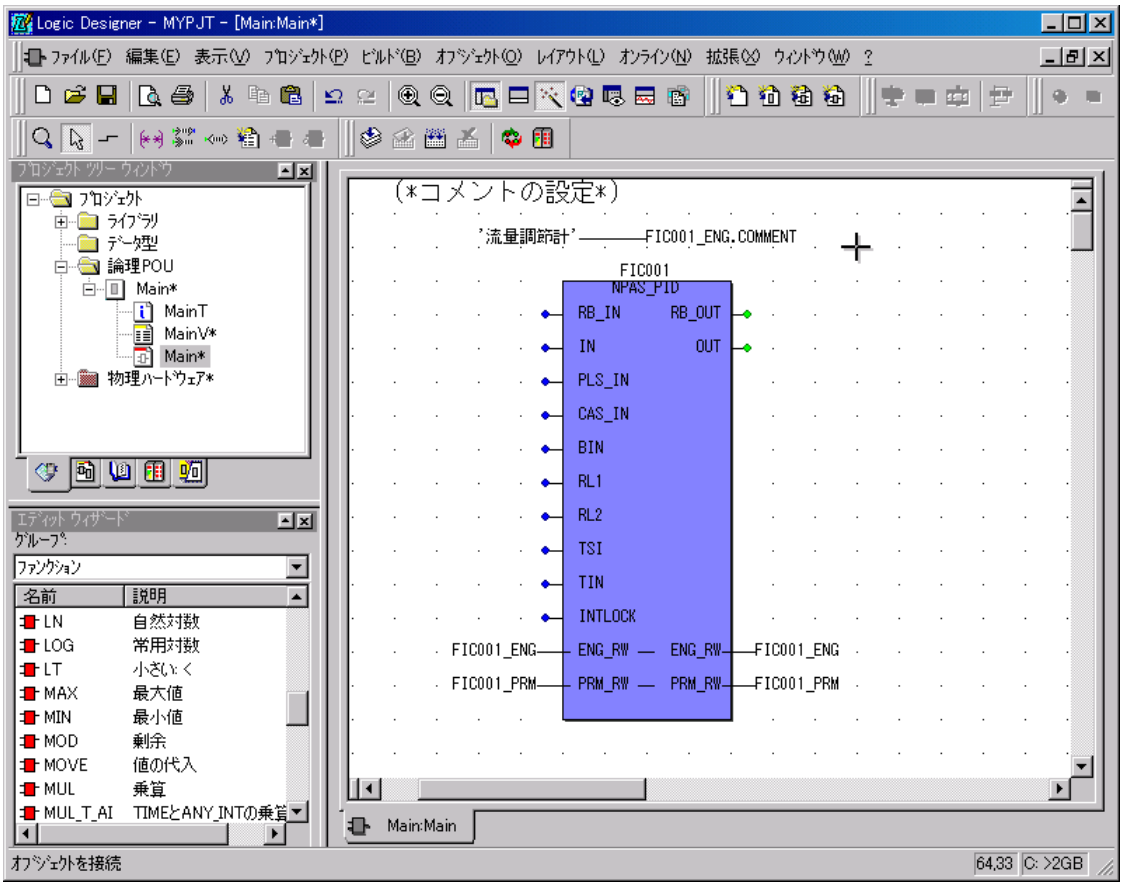


図 エンジニアリングパラメータの設定

■ アクセスパラメータ設定

NPAS_PID のアクセスパラメータを設定する手順を説明します。
先に作成したアクセスパラメータ参照・設定用の変数は，構造体になっており，NPAS POU に存在するすべてのアクセスパラメータをメンバとして持っています。これらのデータを設定するときには，構造体メンバにアクセスする方法で値を設定します。

1. データ設定用のファンクション MOVE を使用します。
2. MOVE の出力端子にアクセスパラメータ参照・設定用変数のメンバを指定します。
   変数プロパティの名前に，先に作成した変数名（例，変数名 FIC001_PRM）を入力し，更に，"."（ドット）を入力すると，アクセスパラメータの一覧が表示されます。この一覧から設定するパラメータを選択します（例えば，上限アラーム設定値を設定する場合は，PH を選択）。

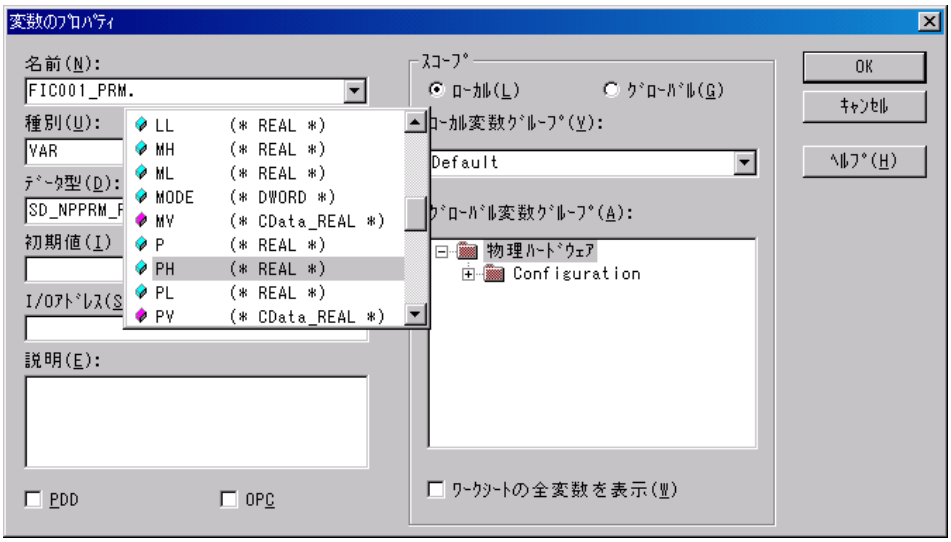


図 アクセスパラメータの選択

3. MOVE の入力端子にアクセスパラメータの設定値となる変数を設定します（例えば，上限アラーム設定値を 70.0m^3/h とする場合は，70.0 を指定）。
4. MOVE の実行スイッチ「EN」に BOOL 型の変数（例，SW01）を設定します。

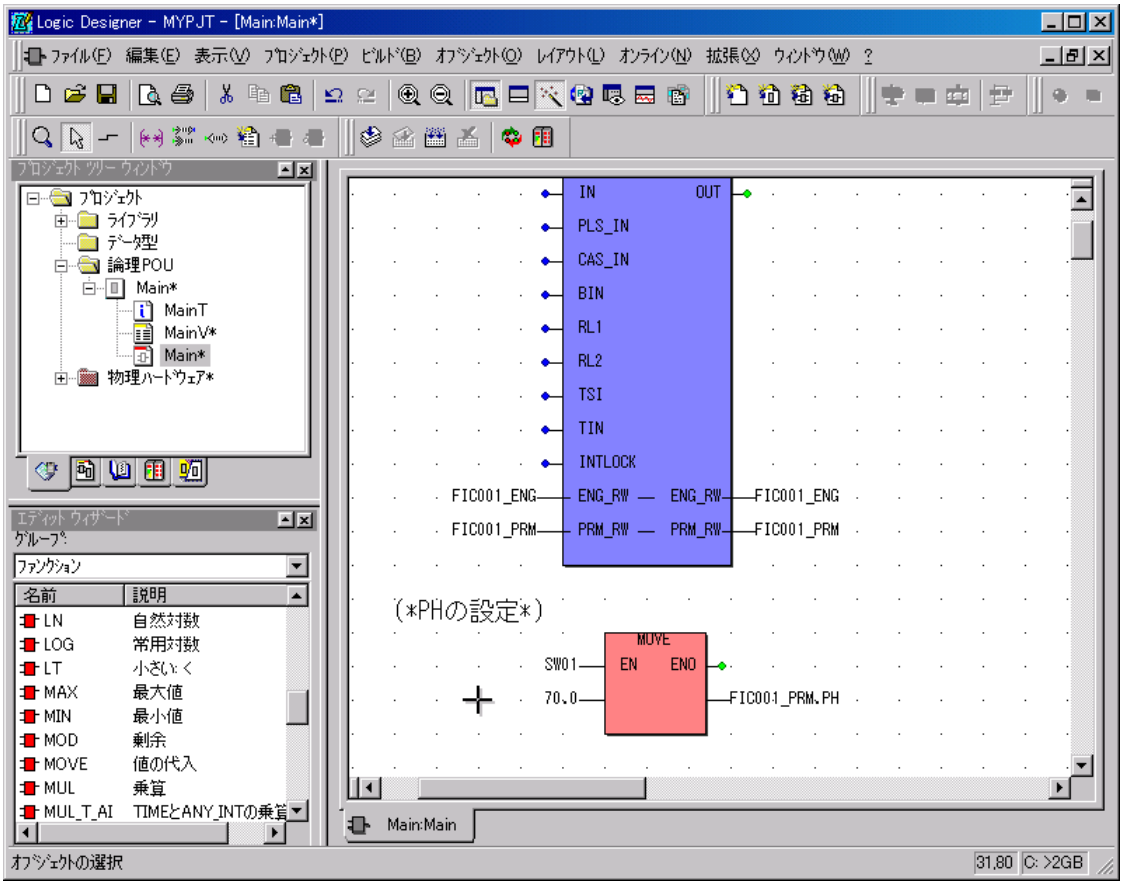


図 アクセスパラメータ PH の設定

NPAS POU へアクセスパラメータが設定される流れは次のようになります。

1. SW01 が TRUE になったときに FIC001_PRM.PH に設定値が設定されます。
2. 設定値は NPAS POU の PRM_RW 端子経由で NPAS POU へ入力されます。
3. NPAS POU の中で設定値が指定されたアクセスパラメータへ設定します。

アクセスパラメータの設定は，スキャンごとにワンショットで行われます。常に MOVE ファンクションの実行スイッチ「EN」を TRUE にしておくと，毎回，指定した値が設定されます（常時設定の場合は，設定値をアクセスパラメータ参照・設定用変数に直結してもかまいません）。通常は，条件判断処理を行い，条件が成立した時のみ，実行スイッチ EN を TRUE にしてアクセスパラメータへの設定を行うようなプログラムを作成します。

■ アクセスパラメータ参照

NPAS POU のアクセスパラメータの値を参照する手順を説明します。
アクセスパラメータの参照も，アクセスパラメータ参照・設定用変数の構造体メンバにアクセスする方法で値を参照します。

1. アクセスパラメータ参照・設定用変数のメンバを指定します。
   変数プロパティの名前に，先に作成した変数名（例，変数名 FIC001_PRM）を入力し，更に，"."（ドット）を入力すると，アクセスパラメータの一覧が表示されます。この一覧から設定するパラメータを選択します（例，モードを参照する場合，MODE を選択）。

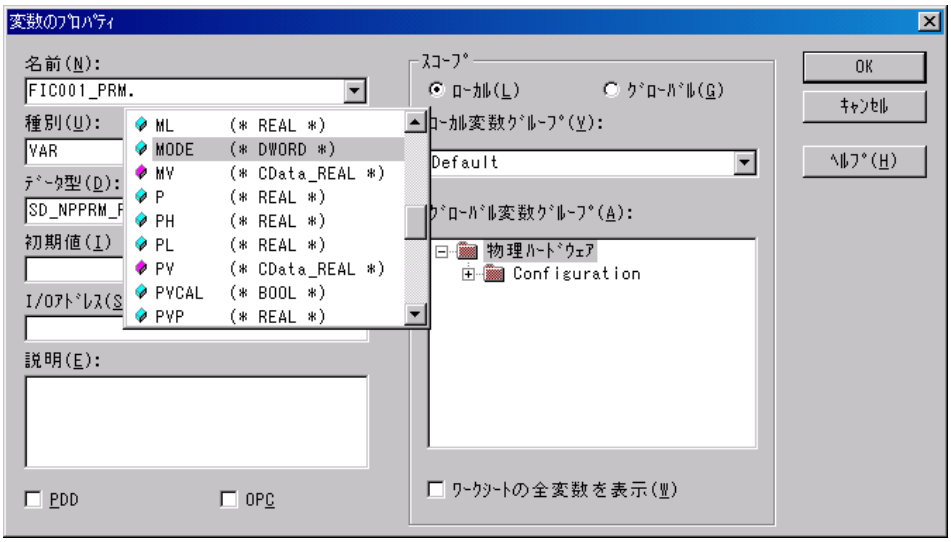


図 アクセスパラメータの選択

2. アクセスパラメータを参照するユーザ定義変数を作成し（例，FIC001_MODE），FIC001_PRM.MODE と接続します。

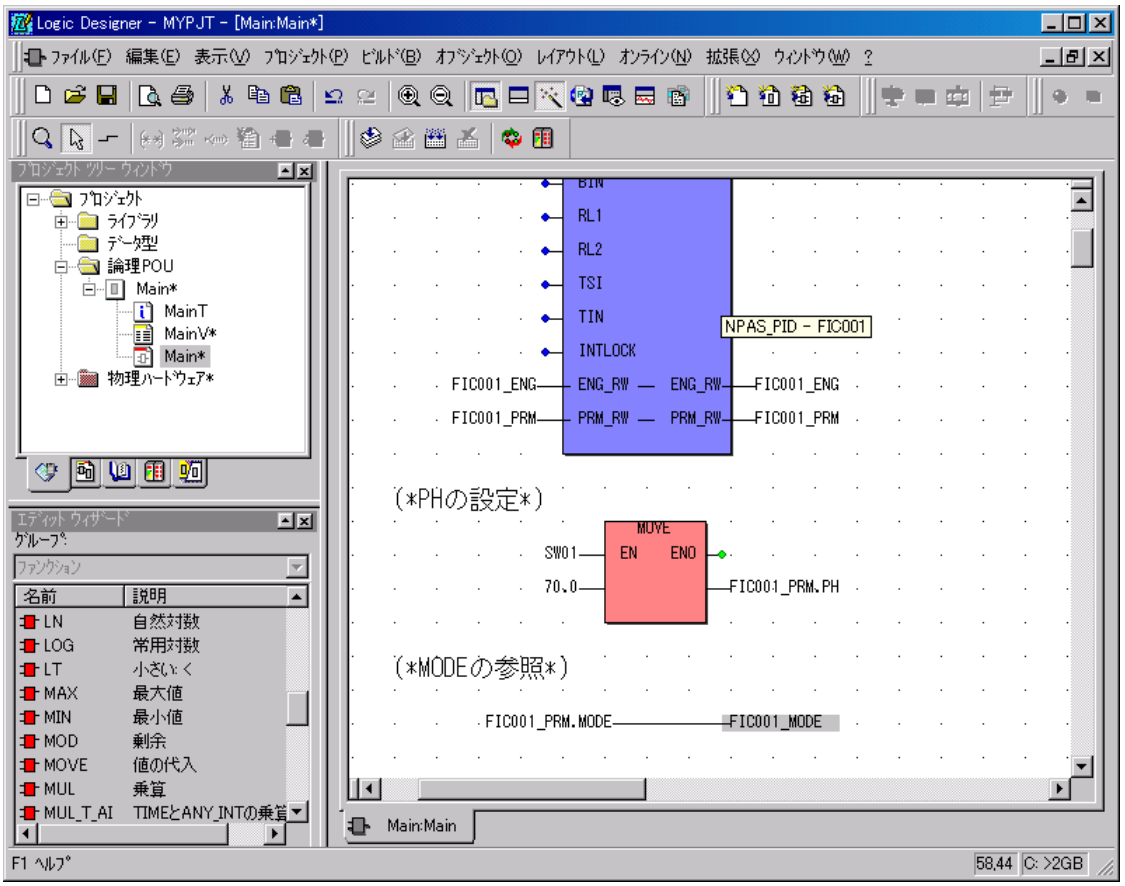


図 アクセスパラメータ MODE の参照

NPAS POU のアクセスパラメータについては，各 NPAS POU の機能詳細にあるアクセスパラメータ一覧を参照してください。

3.3 ロジックデザイナの基本的な操作

本節では，ロジックデザイナの基本的な操作を紹介します。
詳細については，ロジックデザイナのヘルプを参照してください。

■ NPAS POU へのデバイスラベル変数の設定

FCN/FCJ の実 I/O からの入力点・出力点となるデバイスラベル変数は，入力データ処理 POU の IN 端子か，出力データ処理 POU の OUT 端子に設定します。
このデバイスラベル変数の設定方法は，入力データ処理 POU の IN 端子，または，出力データ処理 POU の OUT 端子をダブルクリックし，表示される「変数のプロパティ」ウィンドウで，以下の手順により設定します。
なお，ここでは，デバイスラベル変数の定義は，「物理ハードウェア」－「FCX01」－「DeviceLabelDefinition」で定義済みとして説明します。

1. 「グローバル変数グループ」のところで，「Configuration」の前の「＋」を選択。
2. 「グローバル変数グループ」のところに表示されるリソース名（デフォルトでは「FCX01」）の前の「＋」を選択。
3. 「グローバル変数グループ」のところに表示される「DeviceLavel_Input_A」（アナログ入力の場合）を選択。
4. 「スコープ」のラジオボタンで「グローバル」を選択。
5. 「名前」のコンボボックスでリストを表示し，デバイスラベル変数を指定。
6. 最後に OK ボタンをクリックするとデバイスラベル変数が設定できます。

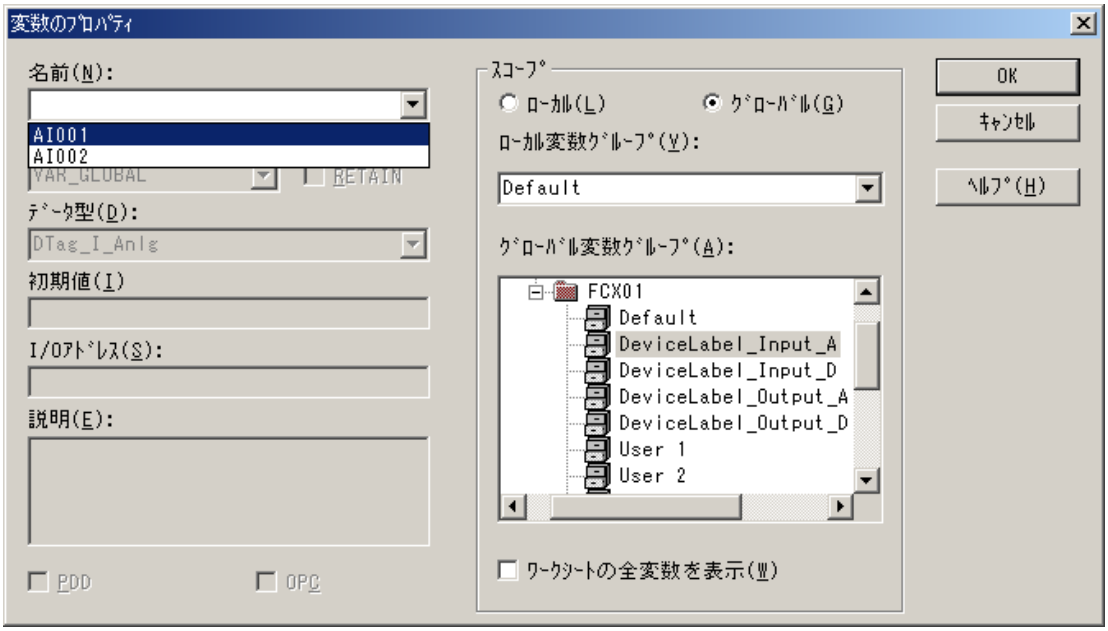


図 デバイスラベル変数の設定

■ ローカル変数の新規作成と設定

NPAS POU では，読み返しデータの設定などで，変数を経由してデータを引き渡す必要があります。この場合は，ワークシート内のみで使用できるローカル変数を作成して行います。
以下に出力データ処理 POU の RB_OUT 端子から連続制御 POU の RB_IN 端子の接続を例にローカル変数の新規作成と作成した変数の指定方法を説明します。

● ローカル変数の新規作成方法

出力データ処理 POU の RB_OUT 端子にローカル変数を新規作成する場合は，出力データ処理 POU の RB_OUT 端子をダブルクリックし，表示される「変数のプロパティ」ウィンドウで以下の手順によりローカル変数の新規作成を行います。

1. 「スコープ」のラジオボタンで「ローカル」を選択。
2. 名前のコンボボックスに，新規に作成するローカル変数名"FIC001_RB"を入力します。
3. 最後に OK ボタンをクリックすることでローカル変数が新規作成され，出力データ処理 POU の RB_OUT 端子に割り付きます。

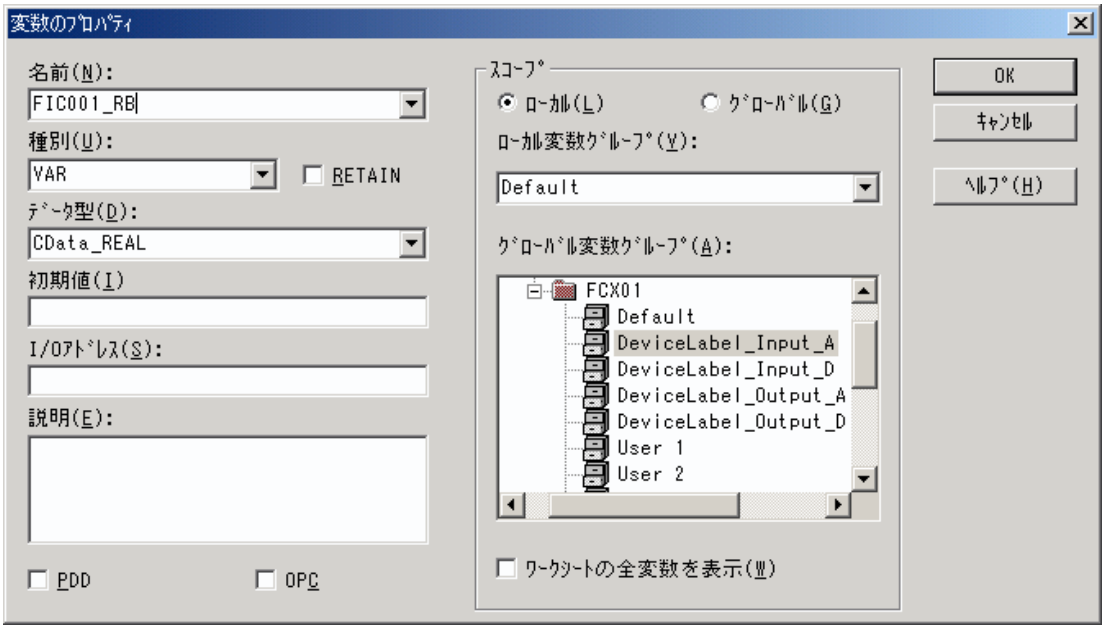


図 ローカル変数の新規作成

● 作成したローカル変数の割り付け方法

連続制御 POU の RB_IN 端子に，先程作成したローカル変数"FIC001_RB"を割り付ける場合は，連続制御 POU の RB_IN 端子をダブルクリックし，表示される「変数プロパティ」ウィンドウで以下の手順によりローカル変数を割り付けます。

1. 「スコープ」のラジオボタンで「ローカル」が選択されていることを確認。
2. 名前のコンボボックスでリストを表示し，FIC001_RB を選択。
3. 最後に OK ボタンをクリックするとローカル変数の指定が RB_IN 端子に割り付きます。

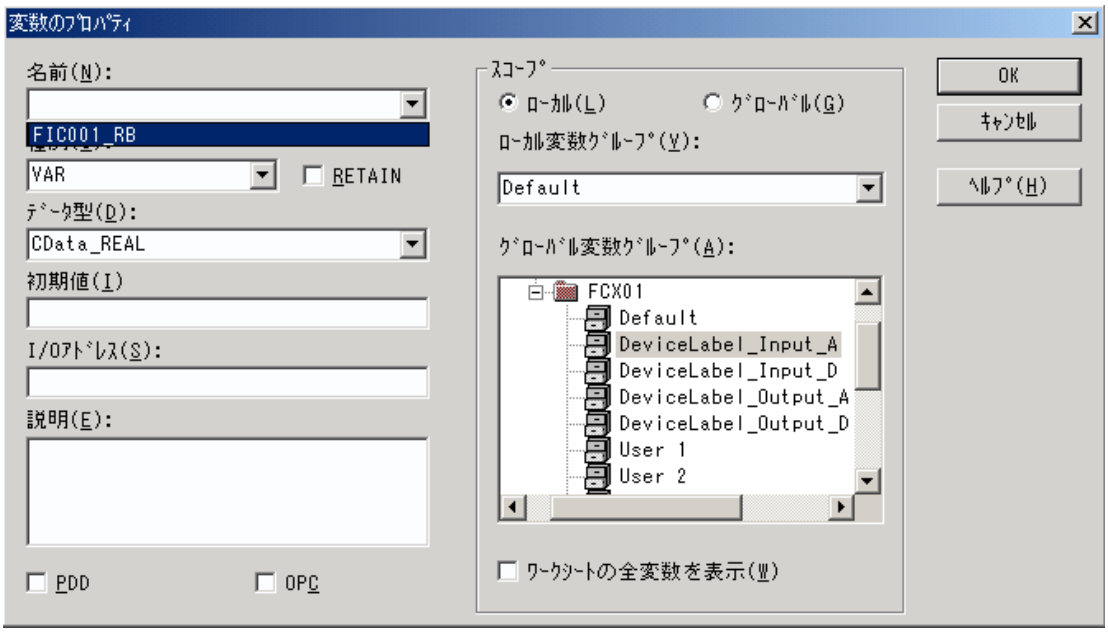


図 ローカル変数の割り付け

■ POU 間の結線

NPAS POU では，POU 間のデータ接続は，POU の出力端子と入力端子を結線することにより行います。
以下に，入力データ処理 POU の OUT 端子と連続制御 POU の IN 端子を結線する方法を説明します。

1.ロジックデザイナのツールバーの「オブジェクトの接続」ボタンを選択します。

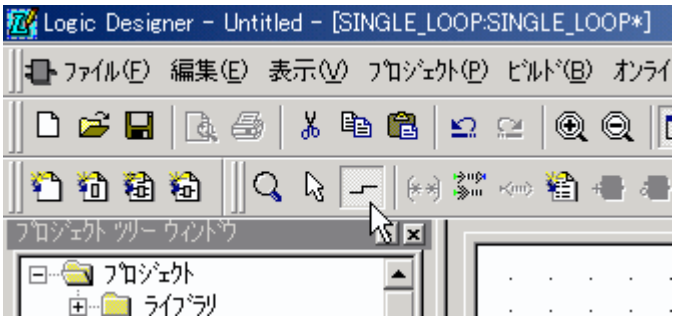


図 「オブジェクトの接続」ボタンの選択

2. 入力データ処理 POU の OUT 端子をクリックします。
3. 入力データ処理 POU の OUT 端子からカーソルまで赤い線が延びるので，そのまま，連続制御 POU の IN 端子をクリックします。

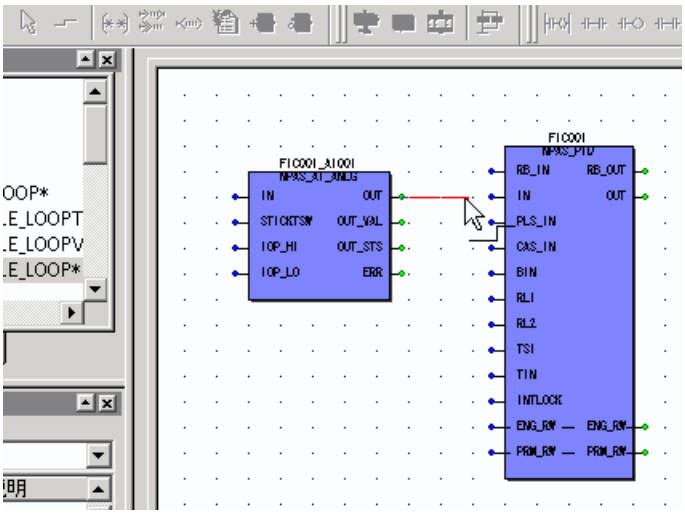


図 NPAS POU 間の結線

以上の操作で，入力データ処理 POU の OUT 端子と連続制御 POU の IN 端子が結線されます。

3.4 プロジェクトのコンパイル，ダウンロードとデバッグ

本節では，プロジェクトのコンパイルとダウンロード，デバッグについて紹介します。

■ タスクへの割り付け

ロジックデザイナで作成したプログラムを実行するためには，プログラムのタスクへの割り付けが必要です。
デフォルトで作成されている"Main"という論理 POU(プログラム)は，デフォルトで"Task0"というタスクに割り付けられていますが，"Main"以外の論理 POU(プログラム)を作成した場合は，作成した論理 POU(プログラム)をタスクに割り付ける必要があります。
以下に，"SINGLE_LOOP"という論理 POU(プログラム)を作成した場合のタスク(Task0)への割り付けの方法について説明します。

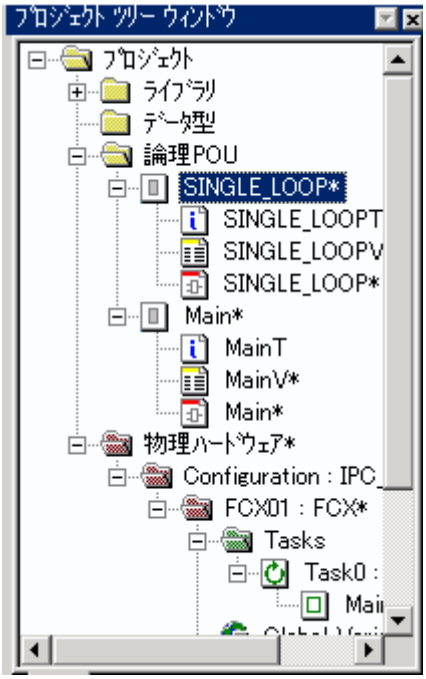


図 新規論理 POU(プログラム) "SINGLE_LOOP"

1. プロジェクトツリーウィンドウの「物理ハードウェア」－「FCX01」－「Tasks」－「Task0」で右クリックし，メニューの「挿入」－「プログラムインスタンス」を選択します。

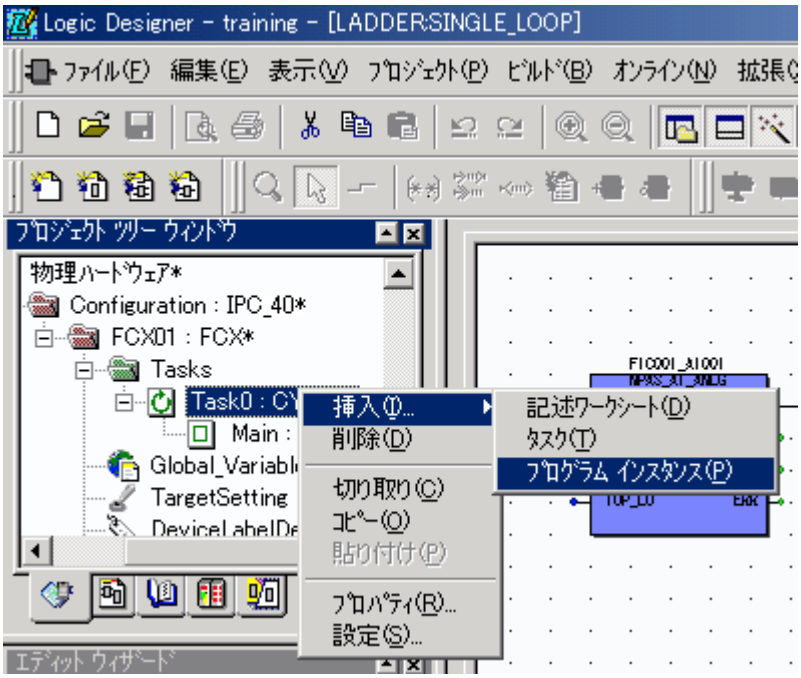


図 タスクの追加

2. 「挿入」のウィンドウで，プログラムタイプのコンボボックスに，新規作成した論理 POU(プログラム)の名前"SINGLE_LOOP"を選択し，プログラムインスタンスに"_SINGLE_LOOP"を入力します。

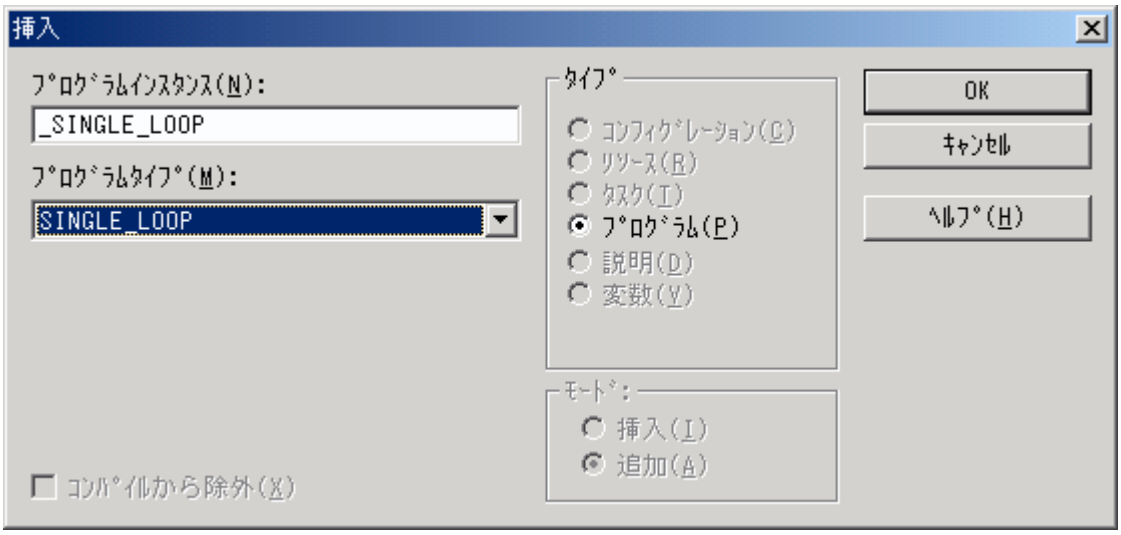


図 「挿入」ウィンドウ

補足

プログラムインスタンスとは，作成した論理 POU(プログラム)を実行するときの名称です。プログラムインスタンスの先頭 1 文字を_（アンダーバー）にすると，制御アプリケーション作成時に定義した変数を VDS/ASTMAC のデータサーバにインポートする際，オブジェクト名からプログラムインスタンスが省略されます。VDS/ASTMAC で，対象論理 POU 上の NPAS POU オブジェクトの操作監視を行う場合は，先頭一文字を_（アンダーバー）にすることを推奨します。

3. 「OK」ボタンをクリックすることで，タスクへの割り付けが行われます。

■ ソフトワイヤリング機能の設定

ロジックデザイナでは，作成したプログラムをデバッグするためにソフトワイヤリングという機能を持っています。
ソフトワイヤリング機能では，制御ロジックの出力と入力をソフトウェアで結合し，実際のプロセス入出力を使用せずに，制御ロジックの動作をデバッグ可能にします。
ソフトワイヤリング機能は，制御ロジックの出力と入力を結合するソフトワイヤリング用のプログラムを作成します。このソフトワイヤリング用プログラムと制御プログラムを FCN/FCJ にダウンロードして，FCN/FCJ 上で同時に動作させることによりデバッグを行います。
以下にソフトワイヤリング機能の設定方法について説明します。

1. プロジェクトツリーウィンドウで，「物理ハードウェア」－「SoftWiring」をダブルクリックします。

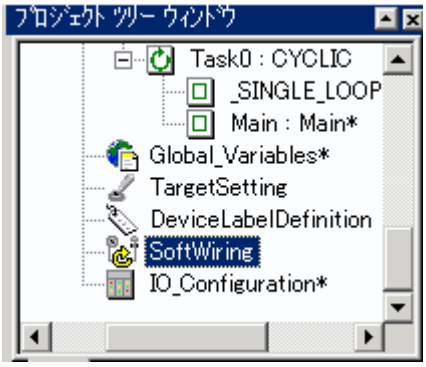


図 プロジェクトツリーウィンドウ

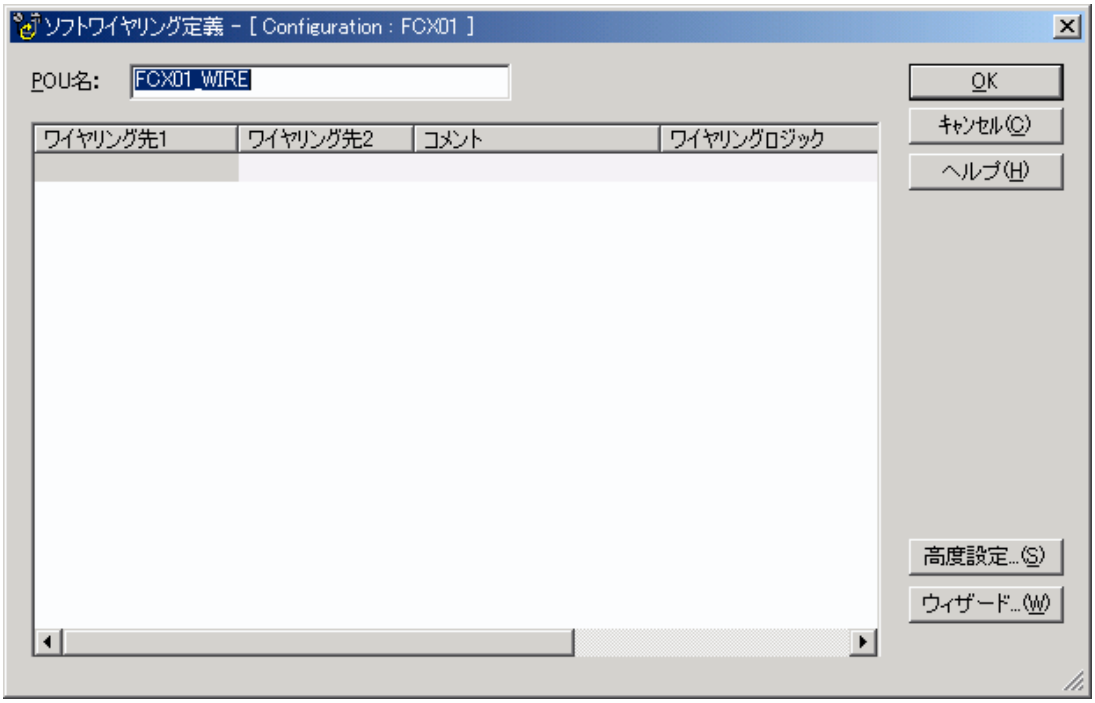


図 ソフトワイヤリング定義ウィンドウ

2. 表示される「ソフトワイヤリング定義」ウィンドウで，「ウィザード」ボタンをクリックします。
3. ワイヤリング先に作成した制御ループの入力のデバイスラベル変数をプルダウンから選択し，「次へ」ボタンをクリックします。
4. 入力形式に「閉ループ型」を選択し，「次へ」ボタンをクリックします。
5. ワイヤリング元に作成した制御ループの出力のデバイスラベル変数をプルダウンから選択し，「次へ」ボタンをクリックします。
6. ワイヤリングロジックに"SD_WIRE_DTANRG_ANRG"を選択し，「次へ」ボタンをクリックします。

補足

"SD_WIRE_DTANRG_ANRG"は，アナログ出力／アナログ入力のソフトワイヤリングのためのロジックです。ソフトワイヤリングのためのロジックは，このほかにも入出力の形式に応じて様々なものを用意しています。

7. 正逆動作やゲイン，バイアス，一時遅れ定数などのワイヤリングロジックを設定して，「次へ」ボタンをクリックします。
8. 最後に今まで設定した内容が表示されるので，内容を確認して「完了」ボタンをクリックします。

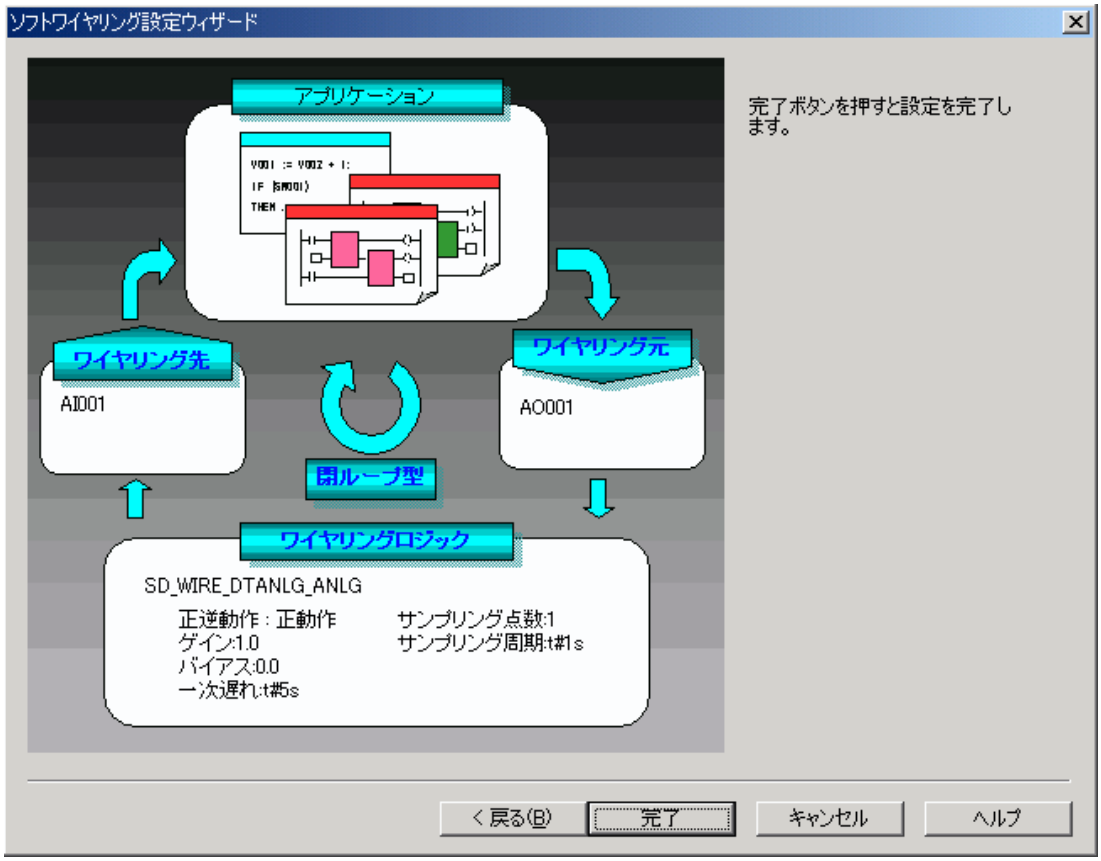


図 設定内容確認画面

9. 「ソフトワイヤリング定義」ウィンドウに戻るので，同じ操作を制御ループの数分繰り返し，最後に「OK」ボタンをクリックします。

10. プロジェクトツリーウィンドウにソフトワイヤリング用プログラムである"FCX01_WIRE"が追加されます。

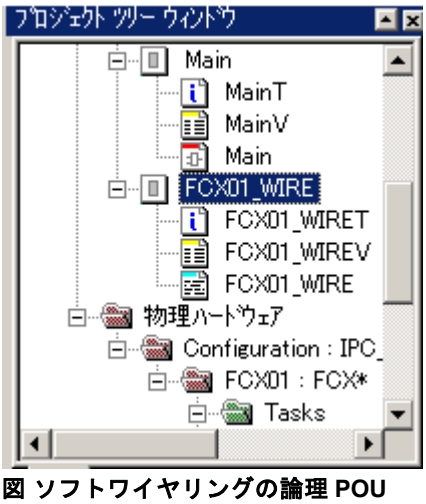


図 ソフトワイヤリングの論理 POU

11.最後にソフトワイヤリング用プログラム"FCX01_WIRE"を Task0 に追加します。
プロジェクトツリーウィンドウの「物理ハードウェア」－「FCX01」－「Tasks」－「Task0」で右クリックし，メニューの「挿入」－「プログラムインスタンス」を選択します。

12. 「挿入」のウィンドウで，プログラムタイプのコンボボックスで，"FCX01_WIRE"を選択し，プログラムインスタンスに"FCX01_WIRE"を入力し，OK ボタンをクリックします。

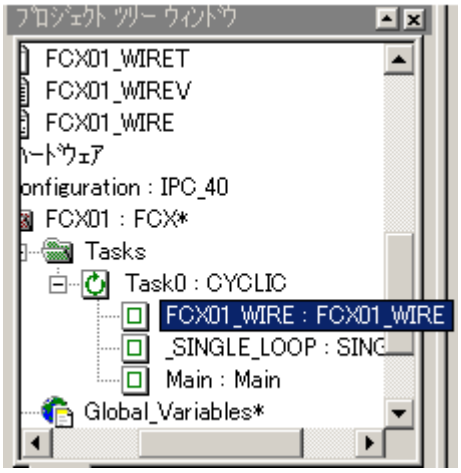


図 "FCX01_WIRE"のタスクへの割り付け

補足

ソフトワイヤリングのプログラムインスタンスには，先頭に_（アンダーバー）をつける必要はありません。

以上で，ソフトワイヤリング機能の設定は終了です。

■ I/O の切り離し（ソフトワイヤリングの動作設定）

ソフトワイヤリングを実際に使用するためには，I/O の切り離しを行う必要があります。
以下に I/O 切り離しの方法について説明します。

1. プロジェクトツリーウィンドウの「物理ハードウェア」－「FCX01」－「Global_Variables」をダブルクリックします。

2. 表示されるリストの項目"GS_NSIO_DISCONF"の初期値を TRUE に変更します。

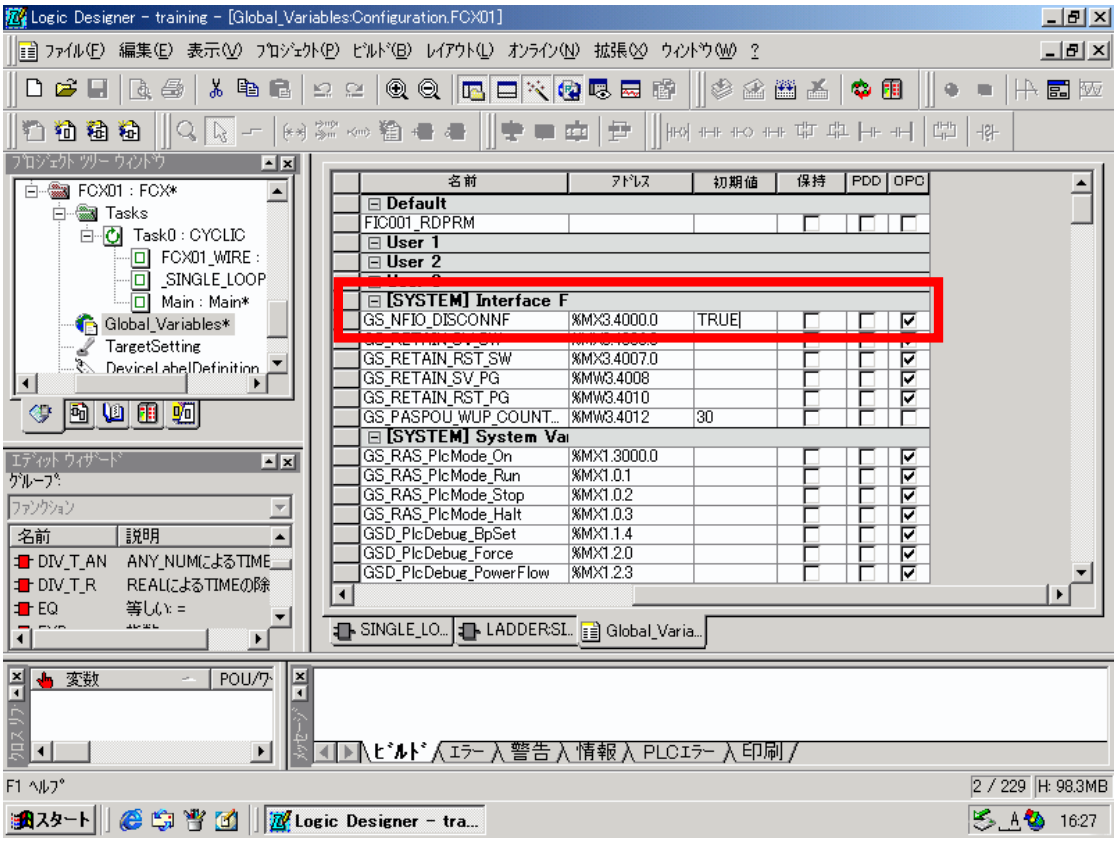


図 I/O 切り離し

以上で，I/O 切り離しの設定は終了です。

重 要

デバッグ終了後，実 I/O に接続して制御アプリケーションを使用する場合は，"GS_NSIO_DISCONF"の初期値を FALSE に変更して，再度プロジェクトのダウンロードを行います。また，ソフトウェア出荷時には，必ずこの変数の初期値を FALSE にしてください。

■ コンパイル

ロジックデザイナでは，作成したプロジェクトを FCN/FCJ 上で実行する形式に変換するためにコンパイルという作業が必要です。
以下にコンパイルの方法について説明します。

1. ロジックデザイナのメニューで「ビルド」－「プロジェクトの再コンパイル」を選択します。

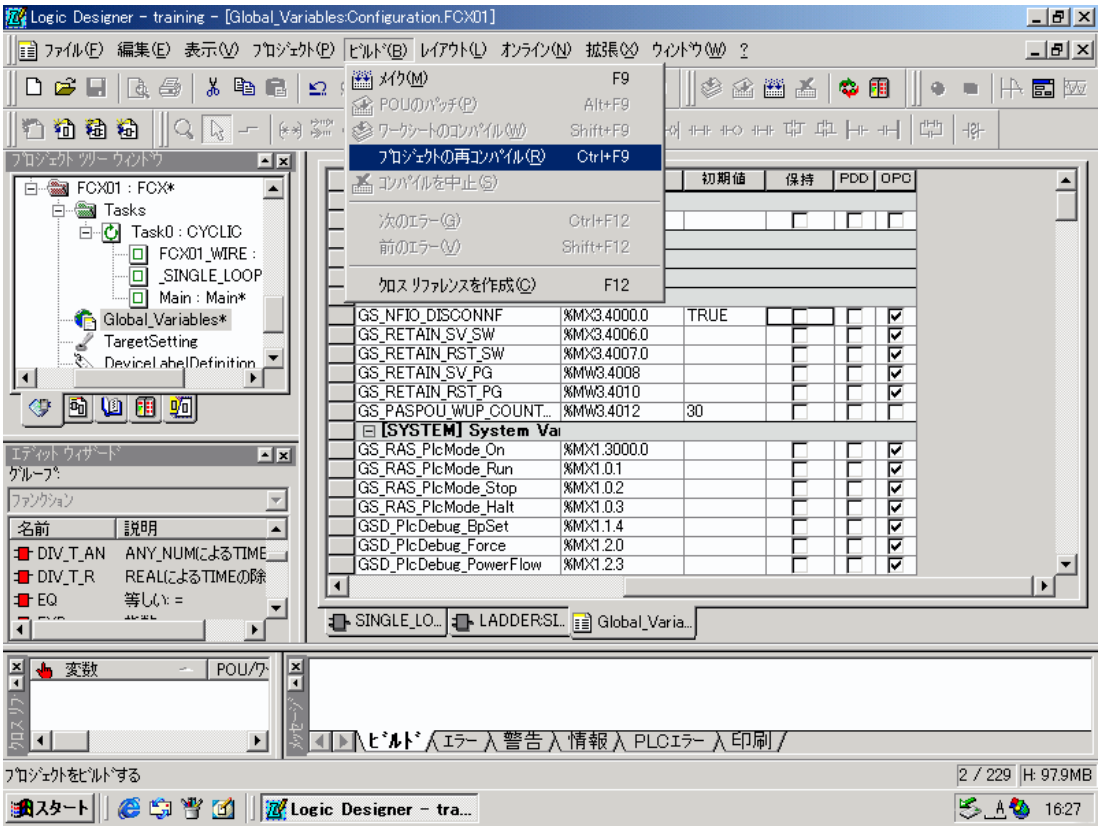


図 プロジェクトの再コンパイル

2. メッセージウィンドウにコンパイル状況が表示されます。最後に「エラー0個」と表示されたら，コンパイルは完了です。

図 メッセージウィンドウ

補足

エラーが発生している場合は，メッセージウィンドウのエラータブを参照し，エラー内容を修正してから，再度コンパイルを行ってください。警告が発生している場合については警告タブを確認して，修正の必要があれば修正してください。タブ上のエラー内容，警告内容をダブルクリックすると対象のワークシートを表示します。

■ ダウンロードとプロジェクトの起動

以下にコンパイルしたプロジェクトの FCN/FCJ へのダウンロードと FCN/FCJ 上でのプロジェクトの起動の方法について説明します。

1. ロジックデザイナのメニューで「オンライン」－「プロジェクトコントロール」を選択すると，プロジェクトコントロールダイアログが表示されます。

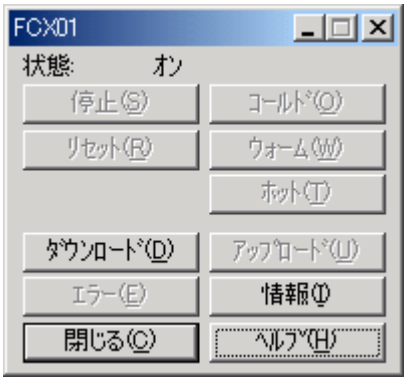


図 プロジェクトコントロールダイアログ

2. プロジェクトコントロールダイアログで，「ダウンロード」ボタンをクリックするとダウンロードウィンドウが表示されます。

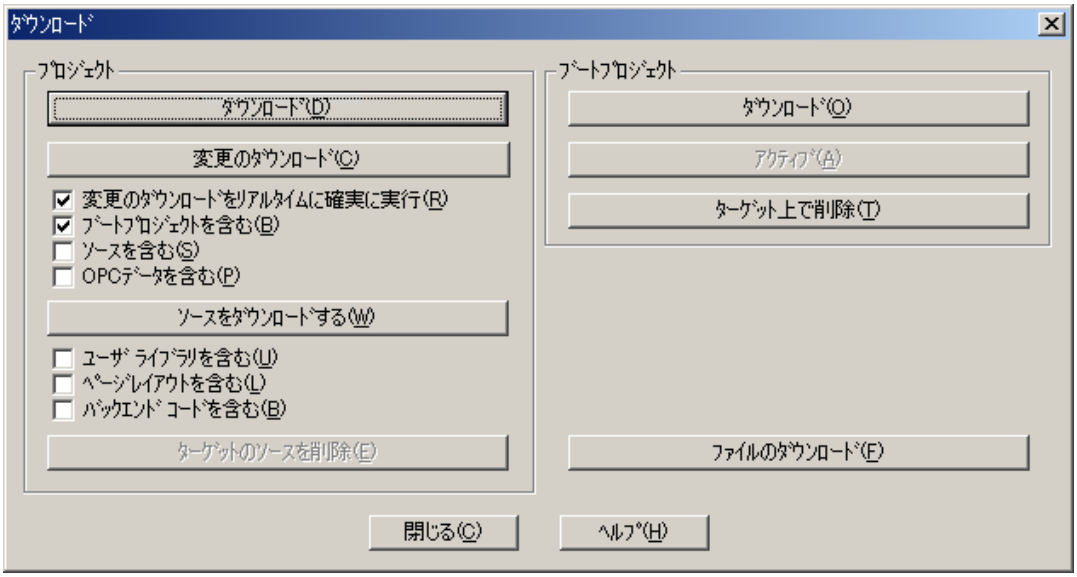


図 ダウンロードウィンドウ

3. ダウンロードウィンドウの「プロジェクト」の「ダウンロード」ボタンをクリックすることで，FCN/FCJ へのダウンロードが開始されます。

補足

図では，「ブートプロジェクトを含む」のチェックボックスがチェックされています。このチェックボックスがチェックされている場合は，揮発性メモリと同時に不揮発性メモリ上にもプロジェクトがダウンロードされます。このチェックボックスがチェックされていない場合は，揮発性メモリのみにダウンロードされるため，FCN/FCJ の電源 OFF でダウンロードされたプロジェクトは消えてしまいます。FCN/FCJ の立ち上げ時には，不揮発性メモリ上のブートプロジェクトが起動するので，特に稼働している FCN/FCJ のエンジニアリング時には，必ずブートプロジェクトもダウンロードしてください。

3. FCN/FCJ へのダウンロードが終了すると，プロジェクトコントロールダイアログの状態が「停止」となります。

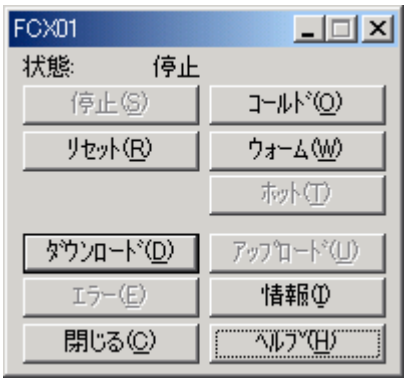


図 プロジェクトコントロールダイアログ

4. プロジェクトコントロールダイアログの「コールド」ボタンをクリックすると状態の表示が「実行」となり，FCN/FCJ 上のプロジェクトが起動します。

補足

FCN/FCJ 上のプロジェクトの起動には，「コールド」，「ウォーム」，「ホット」の 3 種類の方法があり，各々以下の状態でプロジェクトを起動します。
・コールドスタート：非保持データと保持データを初期化してスタート
・ウォームスタート：非保持データを初期化し，保持データを初期化せずにスタート
・ホットスタート：非保持データと保持データとも初期化せずにスタート
ここでの保持データは，「変数のプロパティ」ウィンドウで RETAIN チェックボックスにチェックされている変数のデータを示し，非保持データは，RETAIN チェックボックスがチェックされていない変数のデータを示します。また，NPAS POU のアクセスパラメータも保持データに含まれます。

■ デバッグ

ロジックデザイナには，FCN/FCJ 上のプロジェクトの動作状況を確認し，デバッグを行う機能を持っています。
以下にデバッグするための手順について説明します。

● デバッグモードへの変更

ロジックデザイナで，FCN/FCJ 上のプロジェクトの動作状況を確認するためには，ロジックデザイナをデバッグモードに変更します。
以下にデバッグモードへの変更方法について説明します。

1. ロジックデザイナのツールバーの「デバッグのオン/オフ」アイコンをクリックします。

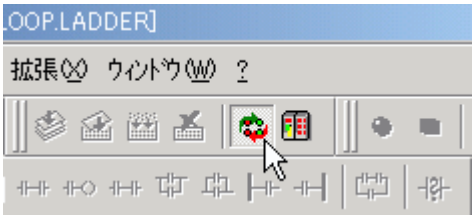


図 「デバッグのオン/オフ」アイコン

以上で，ロジックデザイナのデバッグモードへの変更は終了です。

● ウォッチウィンドウへの変数の割り当て

ロジックデザイナのデバッグモードでは，ウォッチウィンドウへ変数を割り当てることにより，FCN/FCJ 上で動作している POU の変数値を確認することができます。以下にウォッチウィンドウの表示方法と変数の割り当て方法について説明します。

1. ウォッチウィンドウが表示されていない場合は，メニューから「表示」－「ウォッチウィンドウ」を選択し，ウォッチウィンドウを表示させます。

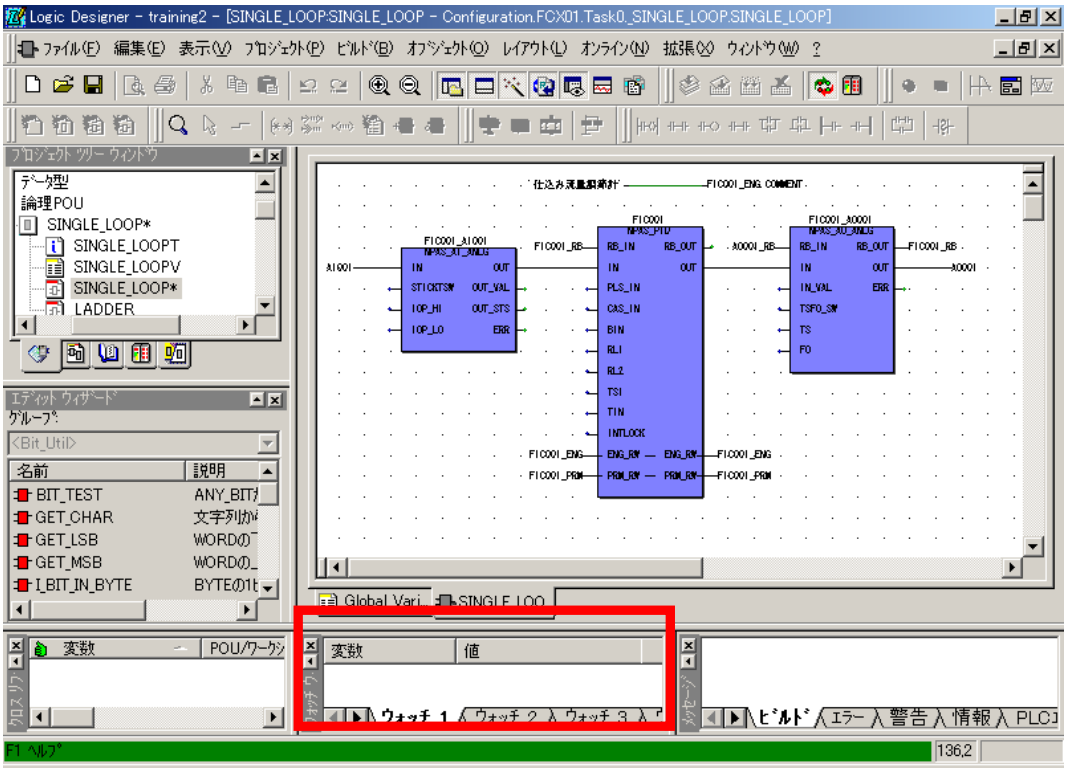


図 ウォッチウィンドウ

2."FIC001"のアクセスパラメータ参照・設定用変数"FIC001_PRM"を右クリックし，メニューの「ウォッチウィンドウに追加する」を選択します。
（ウォッチウィンドウの領域が小さい場合は，ウォッチウィンドウ上で右クリックし，「ドッキング可能」のチェックを外すと独立したウィンドウとして表示されます。）

3. ウォッチウィンドウ上の"FIC001_PRM"を展開すると"FIC001"の各アクセスパラメータの値が表示されます。

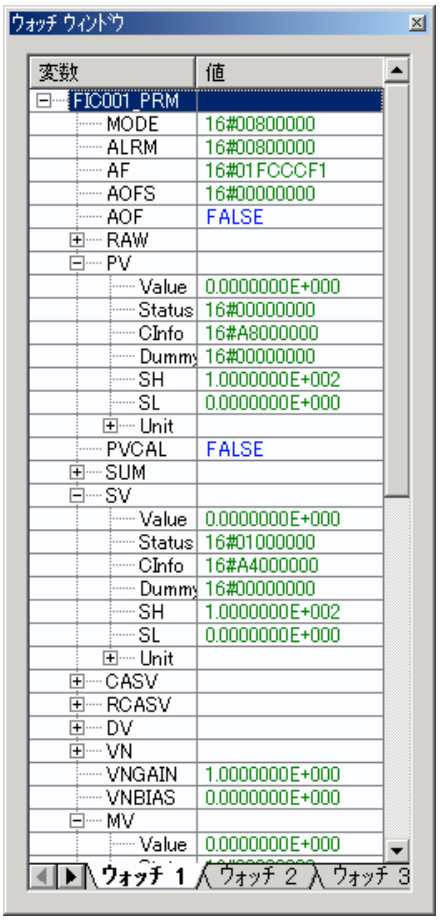


図 独立したウィンドウ表示時のウォッチウィンドウ

● 変数値の変更

ウォッチウィンドウでは，変数値の表示だけではなく，変数値の変更も可能です。以下に変数値の変更方法について説明します。

1. ウォッチウィンドウ上の変更したい変数値，たとえば，SV の Value をダブルクリックします。

2. 変数値を設定するウィンドウが表示されますので，「値」のところに設定したい数値を入力し，「上書き」ボタンをクリックすることで，FCN/FCJ 上のプロジェクトの変数値が設定できます。

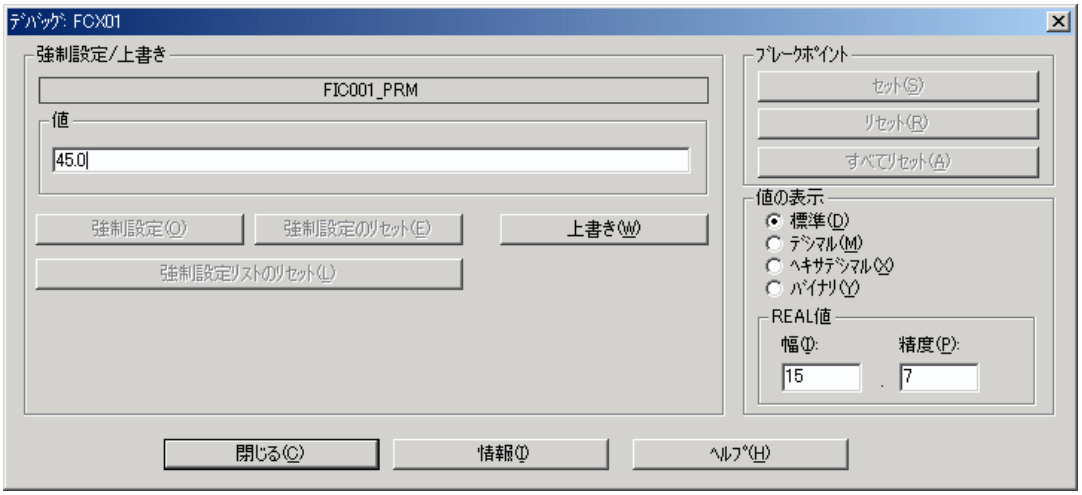


図 変数の変更

補足

REAL 型の変数の変数値を変更する場合は，45.0 のように小数点付きで入力してください。

以上の操作を繰り返し，プロジェクトのデバッグを行います。

3.5 プログラミング上の留意点

本節では，NPAS POU のプログラミング上の留意点について説明します。

■ インスタンス名の文字制限

NPAS POU をインスタンス化するときにつけるインスタンス名の文字数は，通常の変数名の命名規則に従います。実際のインスタンス名は，タスクに割りつけたときのプログラム インスタンス名，各階層に付けられたインスタンス名を含んだものとなります。
このトータルのインスタンス名の文字数の最大文字数が 80 文字です。

● 1 階層の時のインスタンス名

下図の例では，インスタンス名は「PRGNAME.PAS_PID01」です。

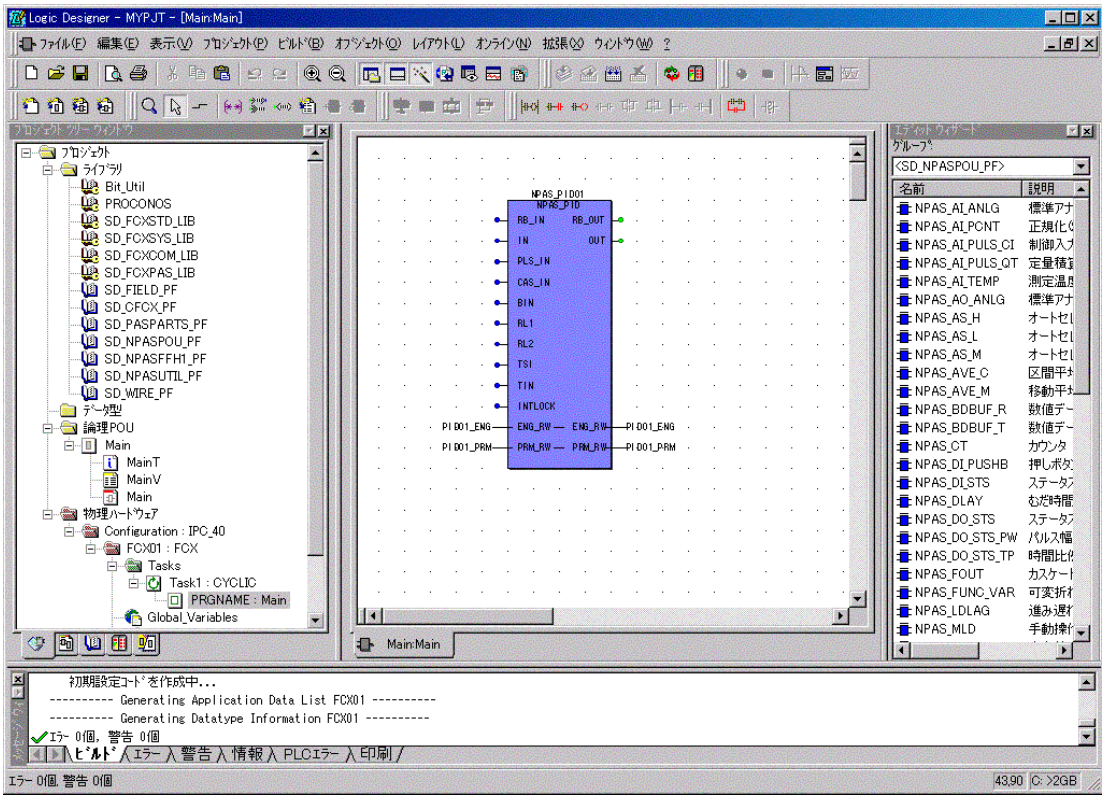


図 1 階層の PAS POU のインスタンス名

● ４階層の時のインスタンス名

NPAS POU を組み合わせて 1 つのまとまった機能をもつファンクションブロックをつくり，それをまた別のファンクションブロックでインスタンス化し・・・というように階層構造にすると，一番下層でインスタンス化されている NPAS POU のインスタンス名は各階層で命名されたインスタンス名を合わせたものとなります。
下図の例では，インスタンス名は
「PRGNAME.CLASS4.CLASS3.CLASS2.CALSS1.PAS_PID01」です。

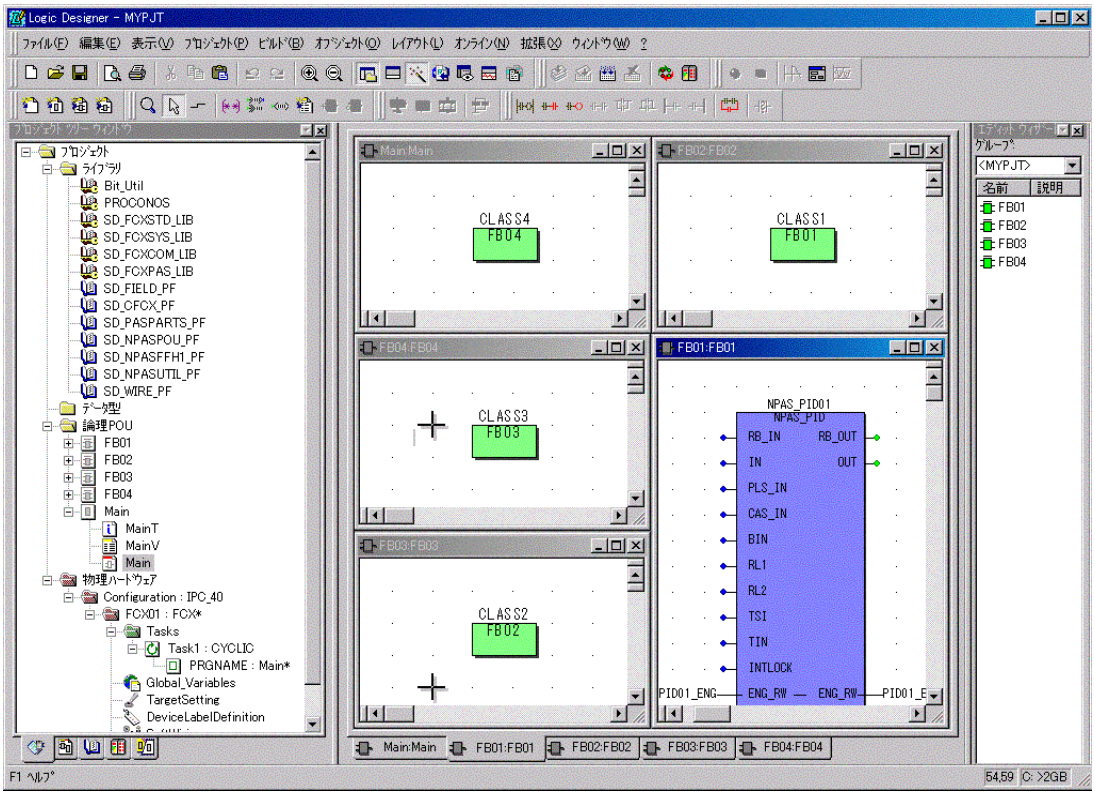


図 4 階層の PAS POU のインスタンス名

● 最大文字数(80 文字)を超えた場合の動作

- NPAS POU の内部で使用するインスタンス名の変数が NULL のままになります。
- インスタンス名が取得できなかったことに対するエラーメッセージなどは発生しません。
- NPAS POU の制御・演算動作には影響がなく，NPAS POU は処理を続行します。
- アラームなどメッセージが発生するような動作が行われた場合には，メッセージ文字列の中にインスタンス名が含まれないため，VDS/ASTMAC はメッセージを正しく処理することができず，ビューアなどにメッセージが表示されません。

■ POU の実行順

NPAS POU が実行される順番について説明します。

● 基本的な実行順

プログラム POU など POU を作成する際にどの言語で記述しても，1 枚のコードワークシートの中で上から下に実行されます。コードワークシートが複数存在する場合は，プロジェクトツリー上の上位にあるシートから実行されます。

● ファンクションブロックダイアグラム(FBD)の実行順

NPAS POU を使用する場合に一番多く使われる言語は，ファンクションブロックダイアグラム(FBD)です。FBD の場合は，基本的に左上から右下に向かって実行されます。結線されている POU 同士は，他の POU の出力を入力している POU が，出力を行っている POU の後に実行されます。

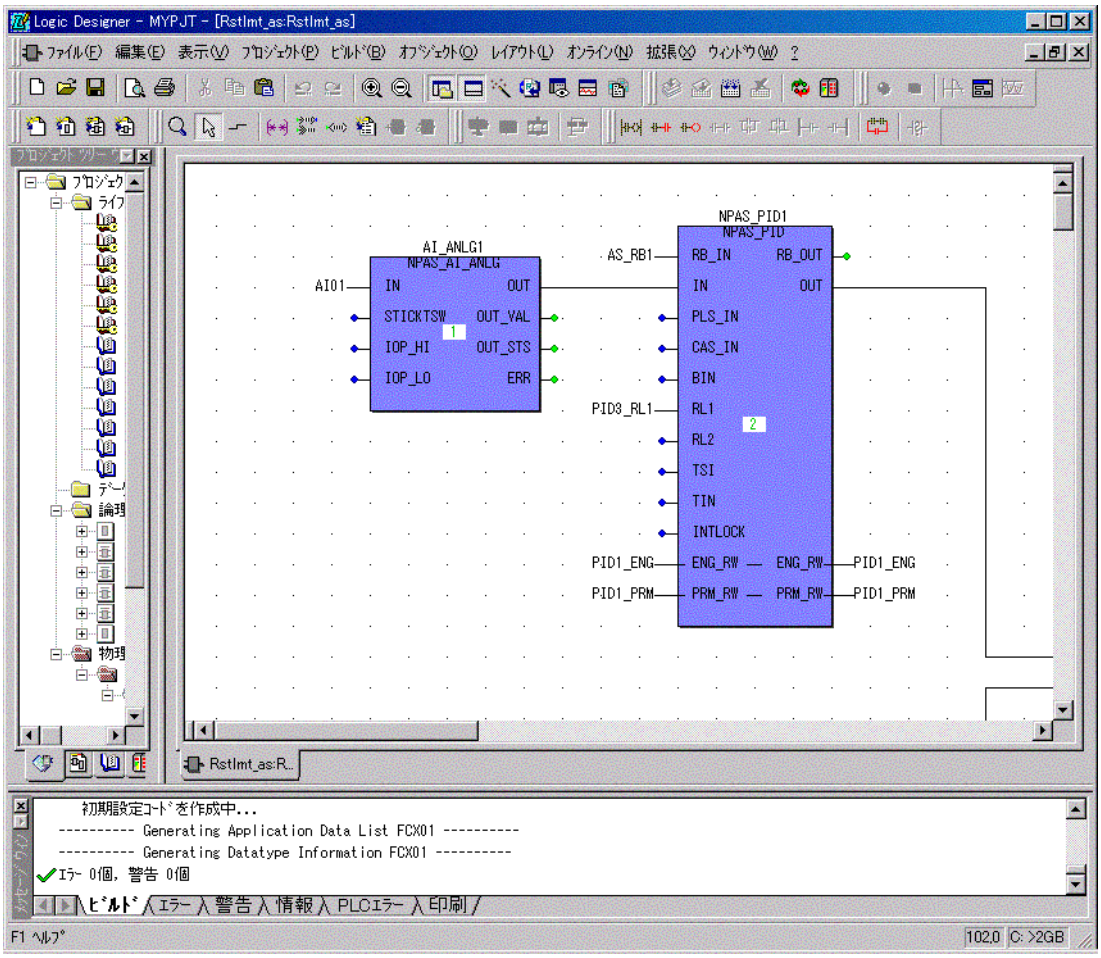


図 FBD の実行順

● NPAS POU の読み返しデータの結合

NPAS POU で必要となる下流の PAS POU からの読み返し(RB_IN)端子への結合は，変数で行うようにすることを強く推奨します。

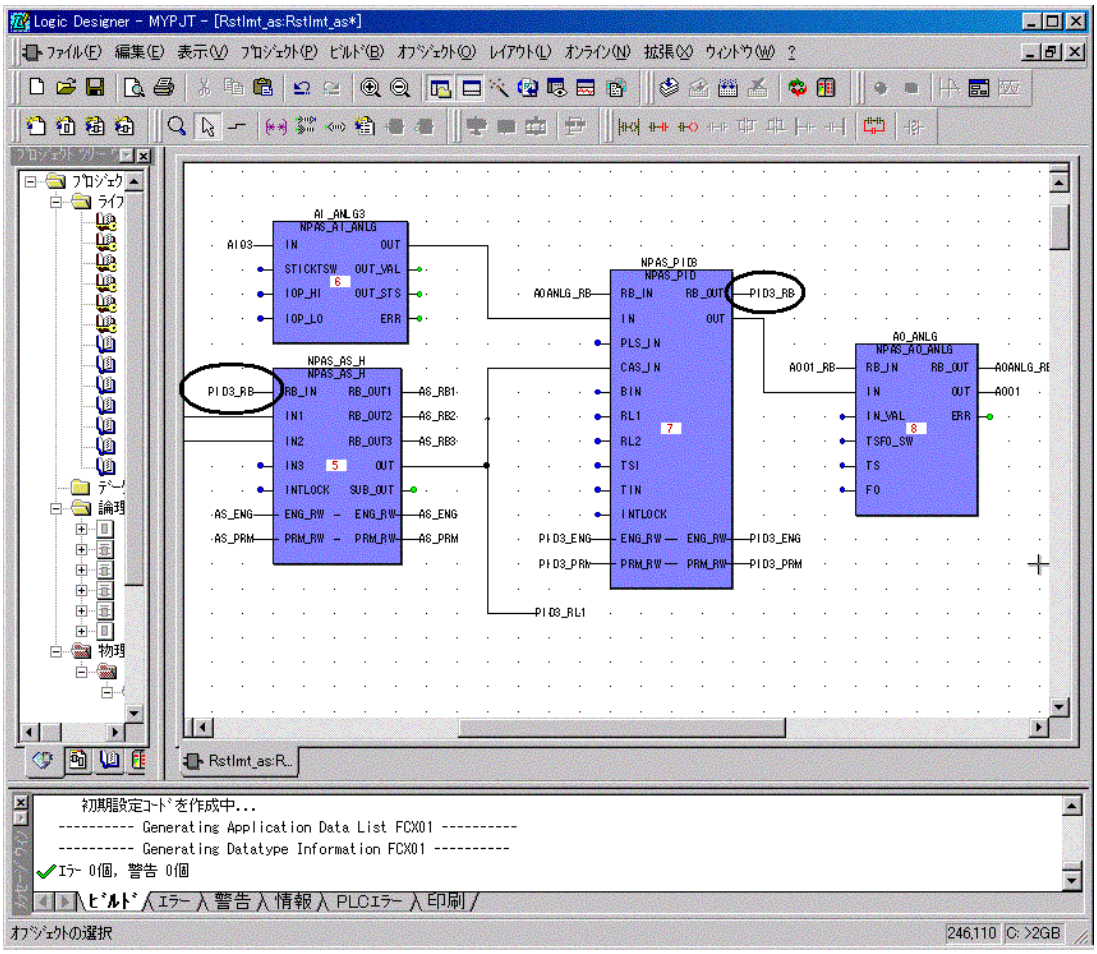


図 FBD の実行順

読み返し(RB_IN)端子への結合を結線で行うと，2 つの POU 間でどちらも相手の出力端子から入力することになるため，実行順序は結線が行われていないときと同様に POU の表示上の位置関係によって決まります。

● 実行順確認の方法

POU の実行順序は，ロジックデザイナの画面で確認することができます。

1. 実行順序を確認したいコードワークシートを表示します。

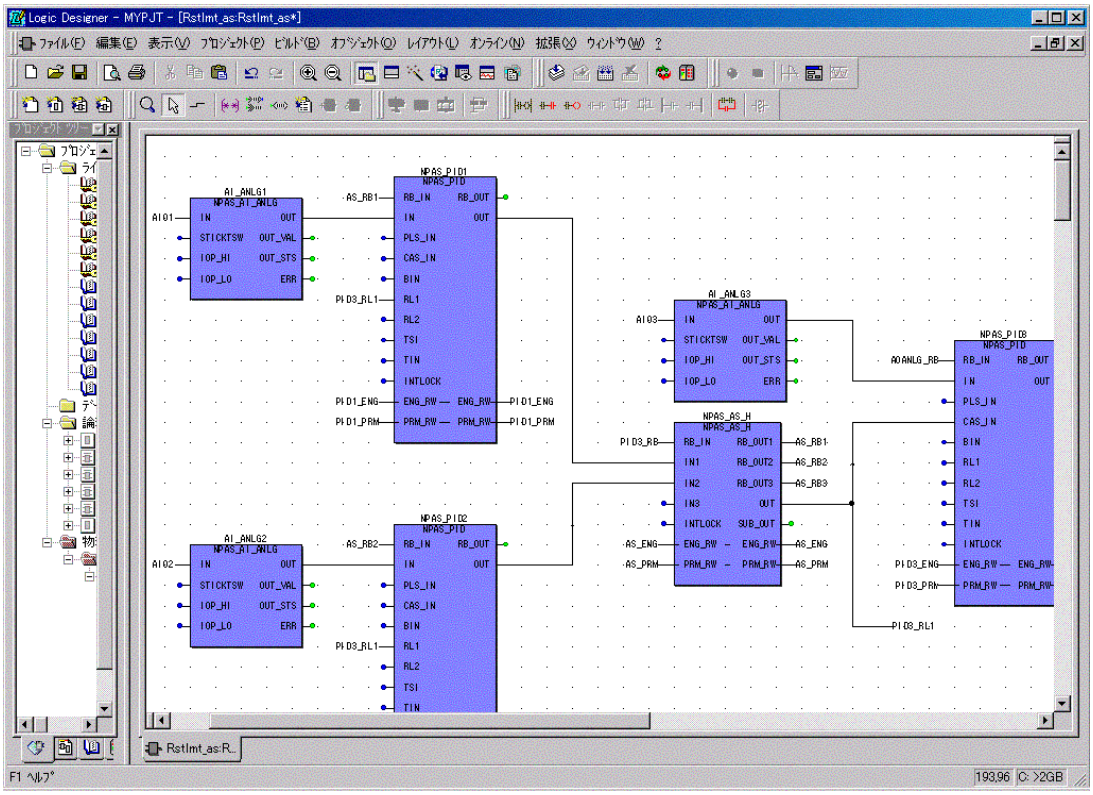


図 コードワークシートの表示

2. コンパイルを実行し，エラーがない状態にしておきます。
3. メニューバーの「レイアウト」－「実行順序」を選択します。

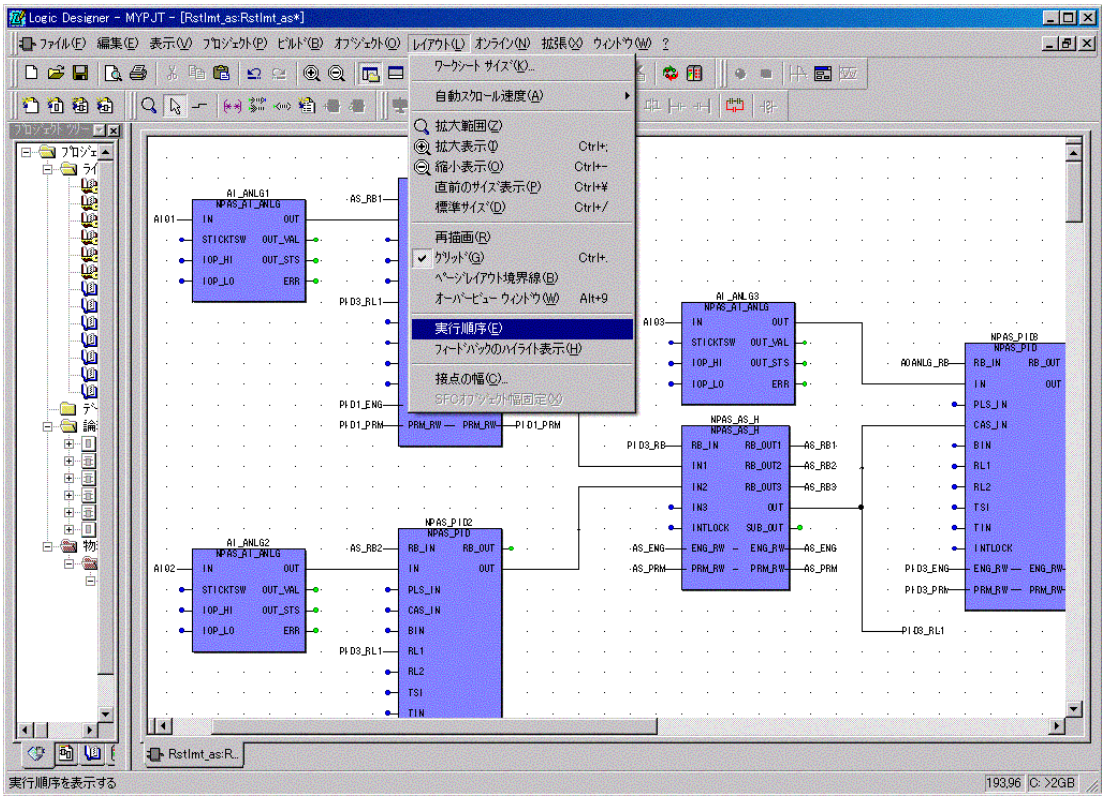


図 メニューからの選択

4. コードワークシートの各 POU に番号が表示されます。この番号が POU の実行される順番になります。

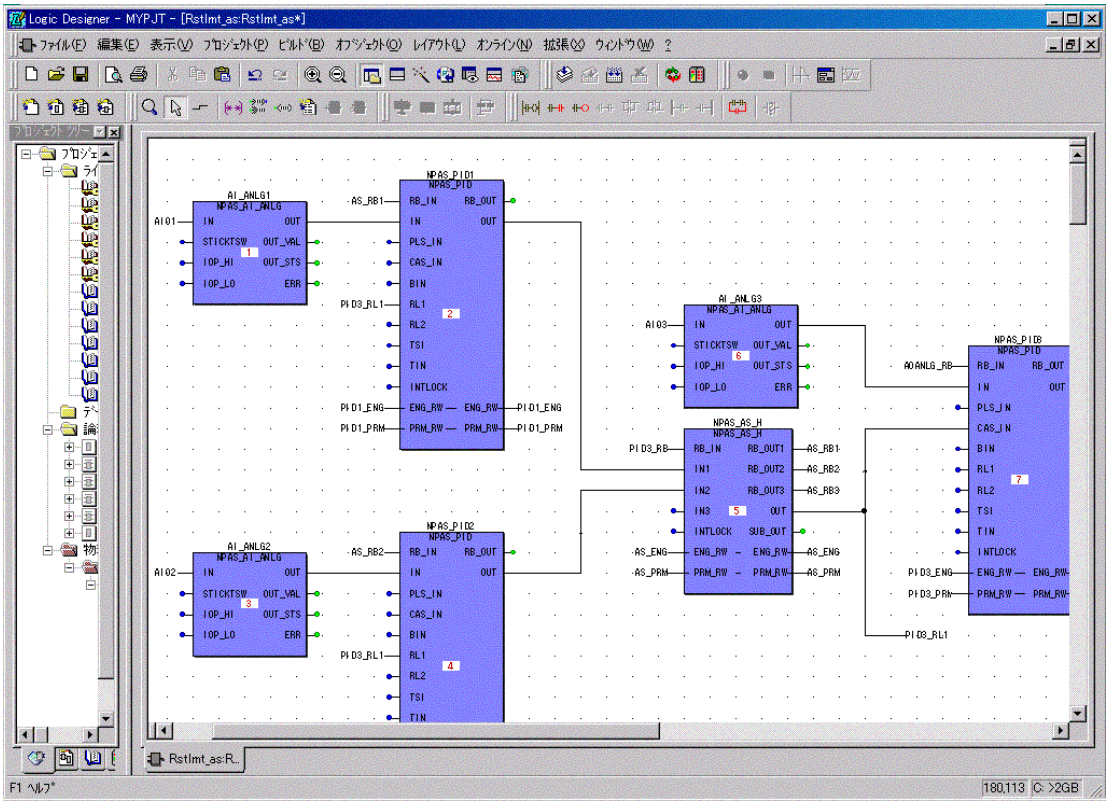


図 実行順の確認

・ 実行順序を表示したまま POU の位置を移動すると，「コンパイルを行ってください」というエラーダイアログが表示されます。コンパイルをかけ直すと，新たに実行順の番号が振り直されます。必要のないときは実行順序を表示しないようにすると，実行順のエラーダイアログは表示されません。

3.6 オンラインダウンロード

オンラインダウンロード機能では，FCN/FCJ の制御動作を継続しながら，制御アプリケーションの変更が行えます。
本節では，NPAS POU のオンラインダウンロード対応について説明します。

■ オンラインダウンロードの可否

NPAS POU の変更作業ごとのオンラインダウンロードの可否を説明します。

表　NPAS POU に関する変更

編集内容	オンライン ダウン ロード可否	備考
POU インスタンス（タグ）の追加	可能	追加された POU インスタンス（タグ）は，初期状態で動作を開始されるため，オンラインダウンロード後にワインドアップ処理を行います。
POU インスタンス（タグ）の削除	可能	
POU インスタンス名（タグ名）の変更	可能	POU インスタンス（タグ）の削除→追加と見なされます。そのため，オンラインダウンロード後に POU インスタンス（タグ）は初期状態で動作を開始します。
POU 型の変更	可能	POU インスタンス（タグ）の削除→追加と見なされます。そのため，オンラインダウンロード後に POU インスタンス（タグ）は初期状態で動作を開始します。
制御入力への新規結合	可能	結合されていなかった制御入力に対して，新規に結合（入力処理 POU など）を定義することを意味します。 この場合，NPAS POU インスタンス（タグ）の動作は，新規結合の結果，制御入力のレンジが変更されるかどうかにより動作が変わります。下記の「制御入力への結合変更」を参照してください。
制御入力への結合変更	可能	制御入力に対する結合を変更することを意味します。 結合変更の結果，制御入力のレンジに変更がない場合は，制御を続行します。結合変更の結果，制御入力のレンジに変更がある場合は，自動運転を停止します(MAN フォールバック)。
制御入力への結合削除	可能	制御入力に対する結合を削除することを意味します。 制御入力がオープンになり，結合削除前の入力値（PV 値）で制御を続行します。
制御出力先の新規結合	可能	結合されていなかった制御出力に対して，新規に結合(出力処理 POU など)を定義することを意味します。 IMAN で出力値（MV 値）をトラッキングした後，制御動作を再開します。
制御出力先の結合変更	可能	制御出力に対する結合を変更することを意味します。 結合変更の結果，出力先のレンジに変更がない場合は，制御を続行します。結合変更の結果，出力先のレンジに変更がある場合は自動運転を停止します(MAN フォールバック)。
制御出力先の結合削除	可能	制御を続行します
エンジニアリングパラメータの変更	可能	制御を続行します

■ オンラインダウンロード実行時の注意

NPAS POU では，複数の POU インスタンス（タグ）が連携して一つの制御ループを形成します。
そのため，オンラインダウンロード時は，修正した POU インスタンスだけでなく，その接続先の POU インスタンスへの入力値の変化についても考慮する必要があります。
また，一部の NPAS POU では，「出力先」の状態も「読み返し入力（"RB_IN 端子"などへの入力）」により入力されていることについても注意する必要があります。

● レンジ変更が発生する入力端子の結合変更

入力端子への結合を変更した結果，POU インスタンス（タグ）への入力データのレンジが変更された場合は，以下のような動作となります。

- 自動運転中（AUT/CAS モード）の POU は，MAN フォールバックとなり，自動運転を停止します。
- 数値演算 POU の一部（NPAS_DLAY，NPAS_LDLAG，NPAS_AVE_M，NPAS_AVE_C）では，内部演算用のデータ（蓄積データ）を初期化します。
- レンジが変更されたアクセスパラメータに依存するアクセスパラメータ（PV に対する HH/LL/PH/PL や，MV に対する MH/ML など）の値は，その工業量が保持されます。
- レンジが変更された結果，各アクセスパラメータ値がスケール外の値となる場合があります。

● ユーザタスク遅延の積算値補正

オンラインダウンロード時にユーザタスクの遅延が発生すると，一般的には積算値に対して影響があります。しかし，R1.70.01 以降の NPAS POU では，自動的に遅延時間を検出して補正します。そのため，NPAS POU 使用時は，ユーザタスク遅延について考慮する必要はありません。

参照

NPAS POU のオンラインダウンロード使用時の注意事項は，上記以外にいくつかあります。詳細は，NPAS POU のオンラインヘルプの「オンラインダウンロードの留意点」の項を参照してください。

■ オンラインダウンロードの手順

オンラインダウンロードの手順について説明します。

● 制御アプリケーションの変更方法

制御アプリケーションの変更は，ロジックデザイナ上で，通常の手順と同様に行います。
このとき，オンラインダウンロード実行可能な変更のみ行います。
制御アプリケーションの変更後，通常の手順と同様にプロジェクトのメイク／再コンパイルを行います。

● オンラインダウンロード実行手順

プロジェクトのメイク／再コンパイルで，エラーがないことを確認したら，プロジェクトコントロールダイアログの「ダウンロード」ボタンを選択します。
表示されるダウンロードダイアログで，「変更のダウンロード」ボタンをクリックすることで，オンラインダウンロードが行われます。

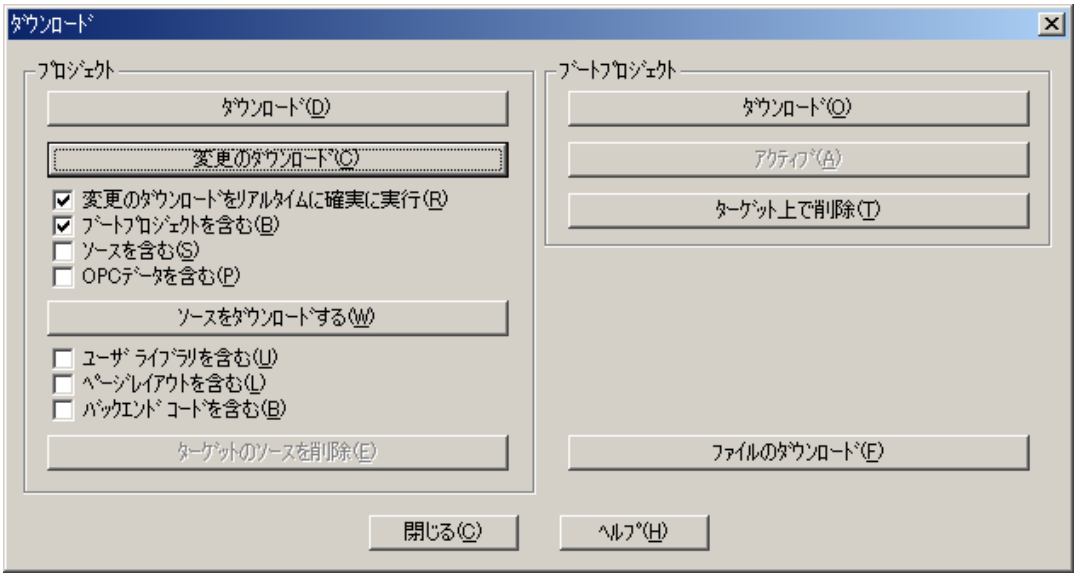


図　ダウンロードダイアログ

補足

オンラインダウンロードを実行できない変更が行われた場合，オンラインダウンロード実行時にエラーとなります。その場合は，コントローラのアプリケーションは変更されず，そのまま継続動作します。そのときには，通常のオフラインダウンロードを実行するか，オンラインダウンロード対象外の変更部分を元に戻してからオンラインダウンロードを実行してください。

Blank Page

付録 A グローバル変数一覧

以下に NPAS POU で使用するシステム用グローバル変数の一覧を示します。

システム用グローバル変数には，定数として使用するグローバル変数とユーザ設定のためのグローバル変数があります。

■ 定数として使用するグローバル変数

定数として使用するグローバル変数の中には，NPAS POU の内部処理でのみ使用されるものと，PAS POU の MODE に AUT や MAN などをアクセスパラメータ結合で設定するときに使用されるものがあります。

定数として使用するグローバル変数の変数名は，"GM_"から始まります。"GM_"で始まるグローバル変数は定数として使用するためのものです。以下の一覧で Read Only(RO)となっているものについては，初期値の変更や，動作途中での変更を行うと NPAS POU が正常に動作しませんので，変更しない定数として利用してください。

● Status Definition

ブロックモード（MODE）の設定やアラーム状態（ALRM）の設定などで使用するグローバル変数です。このグローバル変数は頻繁に使用されます。

表 グローバル変数 ステータスの定義（1/3）

グローバル変数	型	属性	初期値	内容
GM_ALRM_AFL	DWORD	RO	DWORD#16#80000000	AFL
GM_ALRM_AOF	DWORD	RO	DWORD#16#40000000	AOF
GM_ALRM_AF	DWORD	RO	DWORD#16#20000000	AF
GM_ALRM_CAL	DWORD	RO	DWORD#16#01000000	CAL
GM_ALRM_NR	DWORD	RO	DWORD#16#00800000	NR :Normal
GM_ALRM_OOP	DWORD	RO	DWORD#16#00400000	OOP :Output Open Alarm
GM_ALRM_IOPP	DWORD	RO	DWORD#16#00200000	IOP＋:High Input Open Alarm
GM_ALRM_IOPM	DWORD	RO	DWORD#16#00100000	IOP－:Low Input Open Alarm
GM_ALRM_HH	DWORD	RO	DWORD#16#00080000	HH :High High Alarm
GM_ALRM_LL	DWORD	RO	DWORD#16#00040000	LL :Low Low Alarm
GM_ALRM_HI	DWORD	RO	DWORD#16#00008000	HI :High Alarm
GM_ALRM_LO	DWORD	RO	DWORD#16#00004000	LO :Low Alarm
GM_ALRM_DVP	DWORD	RO	DWORD#16#00000800	DV＋ :Deviation Alarm
GM_ALRM_DVM	DWORD	RO	DWORD#16#00000400	DV－ :Deviation Alarm
GM_ALRM_VELP	DWORD	RO	DWORD#16#00000080	VEL＋:Velocity Alarm ＋
GM_ALRM_VELM	DWORD	RO	DWORD#16#00000040	VEL－:Velocity Alarm －
GM_ALRM_MHI	DWORD	RO	DWORD#16#00000020	MHI :Output High Alarm
GM_ALRM_MLO	DWORD	RO	DWORD#16#00000010	MLO :Output Low Alarm

注：表中の属性 RO は，Read Only 属性を示します。

表 グローバル変数 ステータスの定義（2/3）

グローバル変数	型	属性	初期値	内容
GM_ALRM_CNF	DWORD	RO	DWORD#16#00000001	CNF : Connection Failure Alarm
GM_ALRM_ALL	DWORD	RO	DWORD#16#0fffffff	Default Value of AF
GM_ALRM_PERR	DWORD	RO	DWORD#16#00000800	PERR:Undefined Answerback Alarm
GM_ALRM_ANSP	DWORD	RO	DWORD#16#00000080	ANS＋:ON Answerback Error Alarm
GM_ALRM_ANSM	DWORD	RO	DWORD#16#00000040	ANS－:OFF Answerback Error Alarm
GM_ALRM_A1	DWORD	RO	DWORD#16#00008000	A1 :Digital Indicating Controller Alarm1
GM_ALRM_A2	DWORD	RO	DWORD#16#00004000	A2 :Digital Indicating Controller Alarm2
GM_ALRM_A3	DWORD	RO	DWORD#16#00002000	A3 :Digital Indicating Controller Alarm3
GM_ALRM_A4	DWORD	RO	DWORD#16#00001000	A4 :Digital Indicating Controller Alarm4
GM_ALRM_PD1	DWORD	RO	DWORD#16#00000800	PD1 :Power Demand Alarm1
GM_ALRM_PD2	DWORD	RO	DWORD#16#00000400	PD2 :Power Demand Alarm2
GM_ALRM_PD3	DWORD	RO	DWORD#16#00000200	PD3 :Power Demand Alarm3
GM_ALRM_NPLS	DWORD	RO	DWORD#16#00080000	NPLS:Missing Pulse Alarm
GM_ALRM_LEAK	DWORD	RO	DWORD#16#00002000	LEAK:Leak Alarm
GM_ALRM_BDVP	DWORD	RO	DWORD#16#00000800	BDV+:Cumulative Deviation Alarm +
GM_ALRM_BDVM	DWORD	RO	DWORD#16#00000400	BDV-:Cumulative Deviation Alarm -
GM_ALRM_BEND	DWORD	RO	DWORD#16#00000040	BEND:Batch End Alarm
GM_ALRM_BPRE	DWORD	RO	DWORD#16#00000020	BPRE:Pre-batch Alarm
GM_MODE_OS	DWORD	RO	DWORD#16#80000000	O/S :Out of Service
GM_MODE_IMAN	DWORD	RO	DWORD#16#08000000	IMAN:Initialization MANual
GM_MODE_TRK	DWORD	RO	DWORD#16#04000000	TRK :TRacKing
GM_MODE_MAN	DWORD	RO	DWORD#16#00800000	MAN :MANual
GM_MODE_AUT	DWORD	RO	DWORD#16#00400000	AUT :AUTomatic
GM_MODE_CAS	DWORD	RO	DWORD#16#00200000	CAS :CAScade
GM_MODE_PRD	DWORD	RO	DWORD#16#00100000	PRD :PRimary Direct
GM_MODE_RCAS	DWORD	RO	DWORD#16#00080000	RCAS:Remote CAScade
GM_MODE_ROUT	DWORD	RO	DWORD#16#00040000	ROUT:Remote OUTput
GM_MODEMASK_OS	DWORD	RO	DWORD#16#F0000000	OS Mask
GM_MODEMASK_IMAN	DWORD	RO	DWORD#16#0F000000	IMAN Mask
GM_MODEMASK_MAN	DWORD	RO	DWORD#16#00F00000	MAN Mask
GM_MODEMASK_RCAS	DWORD	RO	DWORD#16#000F0000	RCAS Mask
GM_DSTS_QST	DWORD	RO	DWORD#16#40000000	QST :QueSTionable value
GM_DSTS_PTPF	DWORD	RO	DWORD#16#08000000	PTPF:Path To Process Failed
GM_DSTS_CLPP	DWORD	RO	DWORD#16#04000000	CLP＋:CLamP high
GM_DSTS_CLPM	DWORD	RO	DWORD#16#02000000	CLP－:CLamP Low
GM_DSTS_CND	DWORD	RO	DWORD#16#01000000	CND :CoNDitional
GM_DSTS_CAL	DWORD	RO	DWORD#16#00800000	CAL :CALibration
GM_DSTS_OS	DWORD	RO	DWORD#16#00200000	OS :Out of Service
GM_DSTS_MINT	DWORD	RO	DWORD#16#00080000	MINT:Master INiTialization
GM_DSTS_SINT	DWORD	RO	DWORD#16#00040000	SINT:Server INiTialization
GM_DSTS_NSEL	DWORD	RO	DWORD#16#00020000	NSEL:Non SELection switch
GM_DSTS_PRST	DWORD	RO	DWORD#16#00010000	PRST;Pulse width ReSet
GM_DSTS_BAD	DWORD	RO	DWORD#16#00008000	BAD :BAD value
GM_DSTS_IOMT	DWORD	RO	DWORD#16#00004000	IOMT:Input Output MainTenance
GM_DSTS_IOPP	DWORD	RO	DWORD#16#00002000	IOP＋:Input OPen high
GM_DSTS_IOPM	DWORD	RO	DWORD#16#00001000	IOP－:Input OPen low
GM_DSTS_OOP	DWORD	RO	DWORD#16#00000800	OOP :Output OPen
GM_DSTS_NRDY	DWORD	RO	DWORD#16#00000400	NRDY:PI/O Not ReaDY
GM_DSTS_PFAL	DWORD	RO	DWORD#16#00000200	PFAL:PI/O FAiL
GM_DSTS_LPFL	DWORD	RO	DWORD#16#00000100	LPFL:PI/O Long FaiL
GM_DSTS_SIM	DWORD	RO	DWORD#16#00000080	SIM :SIMulation Mode
GM_DSTS_IOCN	DWORD	RO	DWORD#16#00000040	IOCN:IO CoNnection
GM_DSTS_CERR	DWORD	RO	DWORD#16#00000020	CERR:Configuration ERRor
GM_H1ALF_DF	BYTE	RO	BYTE#16#40	H1 Device Status for Input
GM_H1ALF_WF	BYTE	RO	BYTE#16#04	H1 Device Status for Output

注：表中の属性 RO は，Read Only 属性を示します。

表 グローバル変数 ステータスの定義（3/3）

グローバル変数	型	属性	初期値	内容
GM_BSTS_ANCK	DWORD	RO	DWORD#16#80000000	ANCK
GM_BSTS_OFF	DWORD	RO	DWORD#16#00800000	OFF
GM_BSTS_LOCK	DWORD	RO	DWORD#16#00400000	LOCK
GM_BSTS_SIM	DWORD	RO	DWORD#16#00008000	SIM
GM_BSTS_NR1	DWORD	RO	DWORD#16#00004000	NR1
GM_BSTS_RUN	DWORD	RO	DWORD#16#00008000	RUN
GM_BSTS_STOP	DWORD	RO	DWORD#16#00004000	STOP
GM_BSTS_HOLD	DWORD	RO	DWORD#16#80000000	HOLD
GM_BSTS_PALM	DWORD	RO	DWORD#16#00800000	PRE ALRM
GM_BSTS_CTUP	DWORD	RO	DWORD#16#00400000	CTUP
GM_BSTS_NR2	DWORD	RO	DWORD#16#00200000	NR2
GM_BSTS_STRT	DWORD	RO	DWORD#16#00008000	STRT:Batch Start (ZONE10)
GM_BSTS_IBCH	DWORD	RO	DWORD#16#00004000	IBCH:Initial Batch (ZONE1)
GM_BSTS_STUP	DWORD	RO	DWORD#16#00002000	STUP:Setup (ZONE2)
GM_BSTS_STDY	DWORD	RO	DWORD#16#00001000	STDY:Steady (ZONE3)
GM_BSTS_ERLY	DWORD	RO	DWORD#16#00000800	ERLY:Early (ZONE4)
GM_BSTS_PBCH	DWORD	RO	DWORD#16#00000400	PBCH:Pre-batch (ZONE5)
GM_BSTS_END	DWORD	RO	DWORD#16#00000200	END :Batch End (ZONE7)
GM_BSTS_NCNT	DWORD	RO	DWORD#16#00000100	NCNT:Batch Operation Stop (ZONE0)
GM_BSTS_RSET	DWORD	RO	DWORD#16#00000080	RSET:Reset (ZONE6)
GM_BSTS_EMST	DWORD	RO	DWORD#16#00000040	EMST:Emargency Stop (ZONE8)
GM_BSTS_EEMS	DWORD	RO	DWORD#16#00000020	EEMS:Emergency Stop Complete (ZONE9)
GM_BSTS_RSTR	DWORD	RO	DWORD#16#00000010	RSTR:Restart (ZONE11)

注：表中の属性 RO は，Read Only 属性を示します。

参照

アラームやモードなどの詳細については，NPAS POU のオンラインヘルプの「アラーム処理」や「ブロックモード」の項を参照してください。

● Engineering Constant

エンジニアリングパラメータの設定などで使用するグローバル変数です。

表 グローバル変数 エンジニアリングパラメータ用定数

グローバル変数	型	属性	初期値	内容
GM_ANSCK_CHKOPEN	INT	RO	3	開側のみ
GM_ANSCK_CHKCLOSE	INT	RO	2	閉側のみ
GM_ANSCK_CHKBOTH	INT	RO	1	両方
GM_ANSCK_CHKNON	INT	RO	0	なし
GM_ANSIN_DIRECT	INT	RO	0	アンサバック入力動作 正動作
GM_ANSIN_REVERCE	INT	RO	1	逆動作
GM_INTGL_SECOND	DINT	RO	1	秒を基準にしたときの秒の値
GM_INTGL_MINUTES	DINT	RO	60	秒を基準にしたときの分の値
GM_INTGL_HOURS	DINT	RO	3600	秒を基準にしたときの時間の値
GM_INTGL_DATS	DINT	RO	86400	秒を基準にしたときの日（1 日の値）
GM_GNRL_PVSKOFF	BOOL	RO	FALSE	PV 振り切り（保持）
GM_GNRL_PVSKON	BOOL	RO	TRUE	PV 振り切り（振り切り）
GM_GNRL_VNOFF	INT	RO	0	入出力補償 実行スイッチ（補償なし）
GM_GNRL_VNIN	INT	RO	1	入出力補償 実行スイッチ（入力補償実行）
GM_GNRL_VNOUT	INT	RO	2	入出力補償 実行スイッチ（出力補償実行）
GM_GNRL_INITCT	INT	RW	30	初期化カウンタ（ワインドアップ SCAN 回数）
GM_GNRL_CTL_CYCLE	TIME	RW	TIME#1000ms	制御周期 自動決定値（PID_CTL を除く）

注：表中の属性 RO は，Read Only 属性，RW は Read Write 可能属性を示します。

● Application Flags

PAS POU 同士の結合の情報や内部の動作情報，エラーの定義などで使用するグローバル変数です。

表 グローバル変数 アプリケーションフラグ（1/2）

グローバル変数	型	属性	初期値	内容
GM_CINFO_CNT	DWORD	RO	DWORD#16#80000000	CNT：結合有無 なし＝0
GM_CINFO_TRCN	DWORD	RO	DWORD#16#40000000	TRCN：パラメータ接続
GM_CINFO_APCN	DWORD	RO	DWORD#16#20000000	APCN：アクセスパラメータ接続
GM_CINFO_DTPV	DWORD	RO	DWORD#16#08000000	DTPV：PV 系データ
GM_CINFO_DTSV	DWORD	RO	DWORD#16#04000000	DTSV：SV 系データ
GM_CINFO_DTMV	DWORD	RO	DWORD#16#02000000	DTMV：MV 系データ
GM_CINFO_PWCP	DWORD	RO	DWORD#16#00008000	PWCP：パルス幅出力
GM_CINFO_POUT	DWORD	RO	DWORD#16#00004000	POU TYPE 出力変換 POU
GM_CINFO_PIN	DWORD	RO	DWORD#16#00002000	POU TYPE 入力変換 POU
GM_STAT_STARTINIT	DWORD	RO	DWORD#16#00000001	スタート時初期化
GM_STAT_CALCUINIT	DWORD	RO	DWORD#16#00000002	演算初期化
GM_STAT_IMANCOURSE	DWORD	RO	DWORD#16#00000010	IMAN 系状態
GM_STAT_MANCOURSE	DWORD	RO	DWORD#16#00000020	MAN 系状態
GM_STAT_AUTCOURSE	DWORD	RO	DWORD#16#00000040	AUT 系状態
GM_PIODTYPE_IN_ANLG	WORD	RO	WORD#16#0000	アナログ入力 データ形式 標準アナログ入力
GM_PIODTYPE_IN_TEMP	WORD	RO	WORD#16#0100	測定温度入力
GM_PIODTYPE_IN_PCNT	WORD	RO	WORD#16#0200	%入力
GM_PIODTYPE_IN_PLUS	WORD	RO	WORD#16#0300	パルス列入力
GM_PIODTYPE_IN_H1ALG	WORD	RO	WORD#16#0410	FF－H1 アナログ入力
GM_PIODTYPE_IN_H1VARIANT	WORD	RO	WORD#16#0510	FF－H1 汎用入力
GM_PIODTYPE_IN_REAL	WORD	RO	WORD#16#0600	REAL データ入力
GM_PIODTYPE_OUT_ANLG	WORD	RO	WORD#16#0802	アナログ出力 データ形式 標準アナログ出力
GM_PIODTYPE_OUT_H1ALG	WORD	RO	WORD#16#0912	FF－H1 アナログ出力
GM_PIODTYPE_OUT_H1VARIANT	WORD	RO	WORD#16#0A12	FF－H1 汎用出力
GM_PIODTYPE_OUT_PCNT	WORD	RO	WORD#16#0A02	%出力
GM_PIODTYPE_OUT_REAL	WORD	RO	WORD#16#0C02	REAL データ出力
GM_PIODTYPE_IN_STS	WORD	RO	WORD#16#1001	接点入力 データ形式 ステータス入力
GM_PIODTYPE_IN_PUSHB	WORD	RO	WORD#16#1101	押しボタン入力
GM_PIODTYPE_IN_H1DTL	WORD	RO	WORD#16#1211	FF－H1 接点入力
GM_PIODTYPE_IN_DWORD	WORD	RO	WORD#16#1205	DWORD データ入力
GM_PIODTYPE_IN_DINT	WORD	RO	WORD#16#1304	DINT データ入力
GM_PIODTYPE_IN_UDINT	WORD	RO	WORD#16#1404	UDINT データ入力
GM_PIODTYPE_OUT_STS	WORD	RO	WORD#16#1803	接点出力 データ形式 ステータス出力
GM_PIODTYPE_OUT_H1DTL	WORD	RO	WORD#16#1913	FF－H1 接点出力
GM_PIODTYPE_OUT_DWORD	WORD	RO	WORD#16#1A07	DWORD データ出力
GM_PIODTYPE_OUT_DINT	WORD	RO	WORD#16#1B06	DINT データ出力
GM_PIODTYPE_OUT_UDINT	WORD	RO	WORD#16#1C06	UDINT データ出力

注：表中の属性 RO は，Read Only 属性を示します。

表 グローバル変数 アプリケーションフラグ（2/2）

グローバル変数	型	属性	初期値	内容
GM_PIODTYPE_IN_DINTANLG	WORD	RO	WORD#16#3020	DINT アナログデータ入力
GM_PIODTYPE_IN_UDINTANLG	WORD	RO	WORD#16#3120	UDINT アナログデータ入力
GM_PIODTYPE_IN_WORD	WORD	RO	WORD#16#3220	WORD データ入力
GM_PIODTYPE_IN_BYTE8	WORD	RO	WORD#16#3324	BYTE8 データ入力
GM_PIODTYPE_IN_INT24	WORD	RO	WORD#16#3424	INT14 データ入力
GM_PIODTYPE_OUT_DINTANLG	WORD	RO	WORD#16#3822	DINT アナログデータ出力
GM_PIODTYPE_OUT_UDINTANLG	WORD	RO	WORD#16#3922	UDINT アナログデータ出力
GM_PIODTYPE_OUT_WORD	WORD	RO	WORD#16#3A22	WORD データ出力
GM_PIODTYPE_OUT_BYTE8	WORD	RO	WORD#16#3B26	BYTE8 データ出力
GM_PIODTYPE_OUT_INT24	WORD	RO	WORD#16#3C26	INT14 データ出力
GM_ERRNO_ZERODIVIDE	DWORD	RO	DWORD#16#00000001	PIO で使用：0 割を発生させる要因
GM_ERRNO_BADVALUE	DWORD	RO	DWORD#16#00000002	PIO で使用：バッファオーバーのエラー
GM_ERRNO_PRMERROR	DWORD	RO	DWORD#16#00000003	PIO で使用：指定したパラメータのエラー
GM_ERRNO_ERROR	DWORD	RO	DWORD#16#00000004	PIO で使用：その他のエラー
GM_ERRMSG_ZERODIVIDE	STRING	RO	'ZERO DIVIDE'	エラーメッセージ 0 割りを発生させる原因となる値がある
GM_ERRMSG_BADVALUE	STRING	RO	'BAD VALUE: CHANGED DEFAULT'	異常動作をするような値が指定されている
GM_STR_MODE	STRING180	RO	(*1)	MODE のビットに対応する文字列定義
GM_STR_ALRM1	STRING180	RO	(*2)	ALRM のビットに対応する文字列定義
GM_STR_ALRM2	STRING180	RO	(*3)	ALRM のビットに対応する文字列定義
GM_STR_ALRM3	STRING180	RO	(*4)	ALRM のビットに対応する文字列定義
GM_STR_BSTS1	STRING180	RO	(*5)	ブロックステータスのビットに対応する文字列定義
GM_STR_BSTS2	STRING180	RO	(*6)	ブロックステータスのビットに対応する文字列定義
GM_STR_BSTS3	STRING180	RO	(*7)	ブロックステータスのビットに対応する文字列定義
GM_STR_DSTS	STRING180	RO	(*8)	データステータスのビットに対応する文字列定義
GM_STR_ALRM4	STRING180	RO	(*9)	データステータスのビットに対応する文字列定義
GM_STR_BSTS4	STRING180	RO	(*10)	データステータスのビットに対応する文字列定義

注：表中の属性 RO は，Read Only 属性を示します。
*1:'OS::::IMAN:TRK:::MAN:AUT:CAS:PRD:RCAS:ROUT::::::::::::::::::'
*2:':::::::CAL:NR:OOP:IOP:IOP-:HH:LL:::HI:LO:::DV+:DV-:::VEL+:VEL-:MHI:MLO::::CNF:'
*3:':::::::CAL:NR:OOP:IOP:IOP-:HH:LL:::HI:LO:::PERR::::ANS+:ANS-:MHI:MLO::::CNF:'
*4:':::::::CAL:NR:OOP:IOP:IOP-:HH:LL:::A1:A2:A3:A4:PD1:PD2:PD3::VEL+:VEL-:MHI:MLO::::CNF:'
*5:'ANCK::::::::OFF:LOCK:::::::SIM:NR::::::::::::::'
*6:'ANCK::::::::OFF:LOCK:::::::RUN:STOP::::::::::::::'
*7:'HOLD::::::::PALM:CTUP:NR::::::RUN:STOP::::::::::::::'
*8:'QST:::PTPF:CLP+:CLP:CND:CAL::OS::::::BAD:IOMT:IOP:IOP-:OOP:NRDY:PFAL:LPFL:SIM:IOCN:::MINT:SINT:NSEL:PRST'
*9:':::::::CAL:NR:OOP:IOP:IOP-:NPLS:LL:::HI:LO:LEAK::BDV+:BDV-::::BEND:BPRE:::::CNF:'
*10:':::::::::NR::::::STRT:IBCH:STUP:STDY:ERLY:PBCH:END:NCNT:RSET:EMST:EEMS:RSTR::::::'

補足

Application Flags のグローバル変数については，ユーザが直接使用することはほとんどありません。

■ ユーザ設定のためのグローバル変数

ユーザ設定のためのグローバル変数は，インタフェースフラグと呼び，ユーザがFCN/FCJ の制御機能の動作設定を行うために使用されます。

ユーザ設定のためのグローバル変数の変数名は，"GS_"から始まります。本グローバル変数は，必要に応じて初期値の変更や動作途中での値の変更が可能です。

表 インタフェースフラグ

グローバル変数	型	属性	初期値	内容
GS_NFIO_DISCONNF	BOOL	RW	FALSE	FCN/FCJ の NFIO の切り離しスイッチ
GS_RETAIN_SV_SW	BOOL	RW	FALSE	リテインデータのセーブスイッチ TRUE:セーブ開始 FALSE:キャンセル
GS_RETAIN_RST_SW	BOOL	RW	FALSE	リテインデータのリストアスイッチ TRUE:リストア開始 FALSE:キャンセル
GS_RETAIN_SV_PG	INT	RW	0	リテインデータセーブの進捗(0 - 100%)
GS_RETAIN_RST_PG	INT	RW	0	リテインデータリストアの進捗(0 - 100%)
GS_PASPOU_WUP_COUNTER_INIT	INT	RW	30	PAS POU のワインドアップカウンタ

注：表中の属性 RW は Read Write 可能属性を示します。

付録 B デバイスラベルデータ

デバイスラベル変数で使用するデバイスラベルデータは，入力・出力のタイプごとに異なります。
また，デバイスラベル変数に接続する入出力データ処理 POU も入力・出力のタイプごとに異なります。
以下にデバイスラベルデータと使用する入出力処理 POU の一覧を示します。

● デバイスラベルデータと使用する入出力データ処理 POU

表 入力のデバイスラベルデータと入力データ処理 POU 一覧

入力デバイスラベル		入力データ処理 POU	
データタイプ	構造体データ型	NPAS POU 名	出力データ型
標準アナログ入力	DTag_I_Anlg	NPAS_AI_ANLG	CData_REAL
DINT データアナログ入力	DTag_I_DintAnlg	NPAS_AI_DINT	
UDINT データアナログ入力	DTag_I_UdintAnlg	NPAS_AI_UDINT	
REAL データ入力	DTag_I_Real	NPAS_AI_REAL	
周波数入力	DTag_I_Freq	NPAS_AI_FREQ	
測定温度入力	DTag_I_Temp	NPAS_AI_TEMP	
%入力	DTag_I_Pcnt	NPAS_AI_PCNT	
パルス列入力	DTag_I_PulsL	NPAS_AI_PULS_CI (*1)	
		NPAS_AI_PULS_QT (*2)	
ステータス入力	DTag_I_Sts	NPAS_DI_STS	CData_BOOL
押しボタン入力	DTag_I_PushB	NPAS_DI_PUSHB	
WORD データ入力	DTag_I_Word	NPAS_DI_WORD	
FF-H1 アナログ入力	DTag_I_FFAnlg	NPAS_FFI_ANLG	CData_REAL
FF-H1 ステータス入力	DTag_I_FFSts	NPAS_FFI_STS	CData_BOOL
HART バリアブル入力	DTag_I_HART	NPAS_AI_HART	CData_REAL

*1：パルス列入力信号を制御入力用のアナログ量へ変換する POU です。
*2：パルス列入力信号を積算入力用のアナログ量へ変換する POU です。

表 出力のデバイスラベルデータと出力データ処理 POU 一覧

出力デバイスラベル		出力データ処理 POU	
データタイプ	構造体データ型	NPAS POU 名	入力データ型
標準アナログ出力	DTag_O_Anlg	NPAS_AO_ANLG	CData_REAL
DINT データアナログ出力	DTag_O_DintAnlg	NPAS_AO_DINT	
UDINT データアナログ出力	DTag_O_UdintAnlg	NPAS_AO_UDINT	
REAL データ出力	DTag_O_Real	NPAS_AO_REAL	
%出力	DTag_O_Pcnt	NPAS_AO_PCNT	
ステータス出力	DTag_O_Sts	NPAS_DO_STS	CData_BOOL
		NPAS_DO_STS_PW (*1)	CData_REAL
		NPAS_DO_STS_PW 2(*1)	
		NPAS_DO_STS_TP（*2）	
高分解能パルス幅出力	DTag_O_PulsWH	NPAS_DO_STS_PWH (*3)	
WORD データ出力	DTag_O_Word	NPAS_DO_WORD	CData_BOOL
FF-H1 アナログ出力	DTag_O_FFAnlg	NPAS_FFO_ANLG	CData_REAL
FF-H1 ステータス出力	DTag_O_FFSts	NPAS_FFO_STS	CData_BOOL

*1：アナログ量をパルス幅出力信号へ変換する POU です。
*2：アナログ量をある周期の ON/OFF 出力へ変換する POU です。
*3: アナログ量を高分解能パルス幅出力信号へ変換する POU です。

Blank Page

付録 C ロジックデザイナのFUとFB

**NPAS POU では，ロジックデザイナで用意されている FU（ファンクション）や FB（ファンクションブロック）も使用してプログラミングを行います。
以下にロジックデザイナで用意されているファンクション，ファンクションブロックの一覧を示します。各ファンクション，ファンクションブロックの詳細については，ロジックデザイナのヘルプを参照してください。**

■ IEC 61131-3 のファンクション・ファンクションブロック

IEC61131-3 で用意されるファンクションとファンクションブロックの一覧を示します。

表 型変換ファンクション

ファンクション	説明
*_TO_STRING	各型から STRING への型変換 *部分は，BYTE, DWORD, INT, DINT, REAL, SINT, TIME, USINT, UINT, UDINT, WORD
B_BCD_TO_*	BYTE 型 BCD コードから 各型への型変換，*部分は，DINT, INT, SINT
BCD_TO_DINT	DWORD 型 BCD コードから DINT への型変換
BOOL_TO_*	BOOL から 各型への型変換 *部分は，BYTE, WORD, DWORD, SINT, INT, DINT, USINT, UINT, UDINT, REAL
BYTE_TO_*	BYTE から 各型 への型変換 *部分は，BOOL, WORD, DWORD, SINT, INT, DINT, USINT, UNIT, UDINT, REAL
D_BCD_TO_*	DWORD 型の BCD コードから 各型 への型変換，*部分は，SINT, INT, DINT
DINT_TO_*	DINT から 各型への型変換 *部分は，BCD, B_BCD, W_BCD, D_BCD, BOOL, BYTE, WORD, DWORD, SINT, INT, REAL,LREAL, TIME, USINT, UNIT, UDINT
DWORD_TO_*	DWORD から 各型への型変換 *部分は，BOOL, BYTE, WORD, SINT, INT, DINT, REAL, USINT, UINT, UDINT
INT_TO_*	INT から 各型への型変換 *部分は，B_BCD, W_BCD, D_BCD, BOOL, BYTE, WORD, DWORD, SINT, DINT, REAL, USINT, UINT, UDINT
REAL_TO_*	REAL から 各型への型変換 *部分は，BOOL, BYTE, WORD, DWORD,LREAL, SINT, INT, DINT, USINT, UINT, UDINT
LREAL_TO_*	LREAL から 各型への型変換 *部分は， DINT, UDINT, REAL
SINT_TO_*	SINT から各型への型変換 *部分は，B_BCD, W_BCD, D_BCD, BOOL, BYTE, WORD, DWORD, INT, DINT, REAL, USINT, UINT, UDINT
STRING_TO_*	STRING から各型への型変換 *部分は，BYTE, WORD, DWORD, SINT, INT, DINT, REAL, TIME, USINT, UINT, UDINT
TIME_TO_DINT	TIME から DINT への型変換
TRUNC_*	REAL 型データを小数点以下切り捨てて，各型へ型変換，*部分は，SINT, INT, DINT
USINT_TO_*	USINT から 各型への型変換 *部分は，BOOL, BYTE, WORD, DWORD, SINT, INT, DINT, UINT, UDINT, REAL
UINT_TO_*	UINT から 各型への型変換 *部分は，BOOL, BYTE, WORD, DWORD, SINT, INT, DINT, USINT, UDINT, REAL
UDINT_TO_*	UDINT から 各型への型変換 *部分は，BOOL, BYTE, WORD, DWORD, SINT, INT, DINT, USINT, UINT, REAL,LREAL
W_BCD_TO_*	WORD 型 BCD コードから 各型への型変換，*部分は，SINT, INT, DINT
WORD_TO_*	WORD から各型への型変換 *部分は，BOOL, BYTE, DWORD, SINT, INT, DINT, REAL, USINT, UINT, UDINT

表 数値ファンクション

ファンクション	説明
ABS	絶対値
ACOS	アーク コサイン
ASIN	アーク サイン
ATAN	アーク タンジェント
COS	コサイン
EXP	指数
LN	自然対数
LOG	常用対数
SIN	サイン
SQRT	平方根
TAN	タンジェント

表 算術ファンクション

ファンクション	説明
ADD	ANY_NUM 型の加算
ADD_T_T	TIME 型の加算
DIV	ANY_NUM 型の除算
DIV_T_AN	TIME 型の除算
EXPT	べき乗
MOD	剰余算
MOVE	転送
MUL	ANY_NUM 型の乗算
MUL_T_AN	TIME 型の乗算
NEG	2 の補数
SUB	ANY_NUM 型の減算
SUB_T_T	TIME 型の減算

表 ブールビット演算ファンクション

ファンクション	説明
AND	AND 演算
NOT	補数演算
OR	OR 演算
XOR	XOR 演算

表 ビット文字列ファンクション

ファンクション	説明
ROL	左ローテーション
ROL_*	型指定左ローテーション，*部分は BYTE, WORD, DWORD
ROR	右ローテーション
ROR_*	型指定右ローテーション，*部分は BYTE, WORD, DWORD
SHL	左シフト
SHL_*	型指定左シフト，*部分は BYTE, WORD, DWORD
SHR	右シフト
SHR_*	型指定右シフト，*部分は BYTE, WORD, DWORD

表 選択ファンクション

ファンクション	説明
LIMIT	限定値
LIMIT_*	型指定限定値，*部分は，INT, DINT, SINT, REAL, STRING
MAX	最大値
MAX_*	型指定最大値，*部分は，INT, DINT, SINT, REAL, STRING
MIN	最小値
MIN_*	型指定最小値，*部分は，INT, DINT, SINT, REAL, STRING
SEL	バイナリ選択
SEL_*	型指定バイナリ選択，*部分は，BOOL, BYTE, DINT, INT, SINT, REAL, STRING, WORD, DWORD, TIME

表 比較ファンクション

ファンクション	簡単な説明
EQ	比較 : =
GE	比較 : >=
GT	比較 : >
LE	比較 : <=
LT	比較 : <
NE	比較 : <>

表 文字列ファンクション

ファンクション	説明
CONCAT	拡張可能な連結
DELETE	文字列の削除
EQ_STRING	比較 : = (STRING に対して)
FIND	文字の検索
GE_STRING	比較 : >= (STRING に対して)
GT_STRING	比較 : > (STRING に対して)
INSERT	文字の挿入
LE_STRING	比較 : <= (STRING に対して)
LEFT	左端からの文字
LEN	文字列の長さ
LT_STRING	比較 : < (STRING に対して)
MID	中間からの文字
NE_STRING	比較 : <> (STRING に対して)
REPLACE	文字の置き換え
RIGHT	右端からの文字

表 バイステーブル ファンクション ブロック

ファンクション ブロック	説明
SR	優先権のセット
RS	優先権のリセット

表 エッジ検出型ファンクション ブロック

ファンクション ブロック	説明
F_TRIG	立ち下がりエッジ検出
R_TRIG	立ち上がりエッジ検出

表 カウンタ ファンクション ブロック

ファンクション ブロック	説明
CTU	アップカウンタ
CTD	ダウンカウンタ
CTUD	アップダウンカウンタ

表 タイマ ファンクション ブロック

ファンクション ブロック	説明
TP	パルス
TON	オン ディレイ タイマ
TOF	オフ ディレイ タイマ

■ ビット操作ファンクション

ビット操作ファンクションの一覧を以下に示します。本ファンクションは，ライブラリ"Bit_Util"で提供されます。

表 ビット操作ファンクション

ファンクション	説明
BIT_TEST	ビット文字列の 1 ビットの値の読み出し
GET_CHAR	文字列から の 1 文字の抽出
GET_LSB	ビット文字列の下位バイトの値の読み出し
GET_MSB	ビット文字列の上位バイトの値の読み出し
I_BIT_IN_*	ビット文字列の 1 ビットの反転，*部分は，BYTE, WORD, DWORD
PARITY_*	セット ビット数が偶数か奇数かのチェック，*部分は，BYTE, WORD, DWORD
R_BIT_IN_*	ビット文字列の 1 ビットをリセット，*部分は，BYTE, WORD, DWORD
S_BIT_IN_*	ビット文字列の 1 ビットをセット，*部分は，BYTE, WORD, DWORD
SET_LSB	ビット文字列の下位バイトに値を書き込み
SET_MSB	ビット文字列の上位バイトに値を書き込み
STRING_TO_BUFFER	文字列の文字をバッファにコピー
SWAP	ビット文字列の上位バイトと下位バイトを入れ替え

■ FCN/FCJ 基本ファンクションブロック

FCN/FCJ 基本ファンクションブロックの一覧を以下に示します。本ファンクションブロックは，ライブラリ"SD_FIELD_PF"で提供されます。

表 FCN/FCJ 基本ファンクションブロック

ファンクションブロック	説明
PAS_GET_POUINSTN	POU のインスタンス名を取得
PAS_GET_TASKSCAN	POU を動作させるタスクのスキャン周期を取得
PAS_GETTIME	ローカルタイムを取得
PAS_GETTIME2	ローカルタイムをミリ単位で取得
PAS_MSG_UPRCALM	ユーザが指定した文字列をプロセスアラームとして発行
PAS_MSG_USRALM	ユーザが指定した文字列をアラームとして送信
PAS_MSG_USREVT	ユーザが指定した文字列をイベントとして送信
PAS_START_ACT	ワインドアップカウンタのカウントダウンやアプリケーション実行可能状態フラグの操作
PAS_WUP_CHECK	ワインドアップの状態をチェック
PAS_WUP_CT_INIT	ワインドアップカウンタの初期値を取得
PAS_UNIT_TO_BYTE	工業単位文字列を STRING 型から BYTE 型へ変換
PAS_UNIT_TO_STRING	工業単位文字列を BYTE 型から STRING 型へ変換
PAS_OPR_PRM_DWORD	DWORD 型データの各ビットに対する操作
PAS_IS_OLDL_DONE	オンラインダウンロードの実行を検出
PAS_GET_TASKDELAY	オンラインダウンロードの実行によるユーザタスクの遅延時間を取得
SD_CANSDO_ACCESS	SDO アクセスファンクション
SD_FIELD_ACCS_CTRL	アクセスコントロールの変更
SD_FIELD_LANCTRL	Ethernet 動作状態変更
SD_FIELD_LANGETSTS	Ethernet 動作状態取得

■ FCN/FCJ 間通信ファンクションブロック

FCN/FCJ 間通信ファンクションブロックは，FCN/FCJ 間通信で使用するファンクションブロックです。FCN/FCJ 間通信ファンクションブロックの一覧を以下に示します。本ファンクションブロックは，ライブラリ"SD_FCXCOM_LIB"で提供されます。

表 FCN/FCJ 間通信ファンクションブロック

ファンクションブロック	説明	備考
CONNECT	通信相手とのコネクト処理	―
USEND_1V	通信相手にデータを送信(Unconfirmed)	1 回の要求で 1 個のデータを送信
USEND_5V	通信相手にデータを送信(Unconfirmed)	1 回の要求で 5 個のデータを送信
USEND_10V	通信相手にデータを送信(Unconfirmed)	1 回の要求で 10 個のデータを送信
URECV_1V	通信相手からのデータを受信(Unconfirmed)	1 回の要求で 1 個のデータを受信
URECV_5V	通信相手からのデータを受信(Unconfirmed)	1 回の要求で 5 個のデータを受信
URECV_10V	通信相手からのデータを受信(Unconfirmed)	1 回の要求で 10 個のデータを受信
WRITE_1V	通信相手にデータを送信(Confirmed)	1 回の要求で 1 個のデータを送信
WRITE_5V	通信相手にデータを送信(Confirmed)	1 回の要求で 5 個のデータを送信
WRITE_10V	通信相手にデータを送信(Confirmed)	1 回の要求で 10 個のデータを送信
READ_1V	通信相手からのデータを受信(Confirmed)	1 回の要求で 1 個のデータを受信
READ_5V	通信相手からのデータを受信(Confirmed)	1 回の要求で 5 個のデータを受信
READ_10V	通信相手からのデータを受信(Confirmed)	1 回の要求で 10 個のデータを受信

■ FA-M3 エミュレーション(シリアル通信)ファンクションブロック

FA-M3 エミュレーション(シリアル通信)ファンクションブロックは，FCN/FCJ の COM ポートと RS-232C 経由で接続された上位コンピュータ（または表示器）から送出される， FA-M3 用のコマンドをエミュレートするファンクションブロックです。FA-M3 エミュレーション(シリアル通信)ファンクションブロックの一覧を以下に示します。本ファンクションブロックは，ライブラリ"SD_CFAM3R_PF"で提供されます。

表 FA-M3 エミュレーション(シリアル通信)ファンクションブロック

ファンクションブロック	説明
SD_CFAM3R_OPEN	FA-M3 エミュレートタスクの起動を行い，RS-232C の通信を開始

■ 通信ユーティリティファンクションブロック

通信ユーティリティファンクションブロックは，各種通信ライブラリで使用する補助機能を提供します。通信ユーティリティファンクションブロックの一覧を以下に示します。本ファンクションブロックは，ライブラリ"SD_CUTIL_PF"で提供されます。

表 通信ユーティリティファンクションブロック

ファンクションブロック	説明
SD_CUTIL_TRIG	通信要求トリガ生成
SD_CUTIL_W2DW	2 つの WORD 型データから DWORD 型へのデータ変換
SD_CUTIL_W2R	2 つの WORD 型データから REAL 型へのデータ変換
SD_CUTIL_DW2W	DWORD 型から 2 つの WORD 型データへのデータ変換
SD_CUTIL_DW2R	DWORD 型から REAL 型へのデータ変換
SD_CUTIL_R2W	REAL 型から 2 つの WORD 型データへのデータ変換
SD_CUTIL_R2DW	REAL 型から DWORD 型へのデータ変換
SD_CUTIL_WORDSWAP	WORD 型データのエンディアン変換
SD_CUTIL_DWORDSWAP	DWORD 型データのエンディアン変換

■ Ethernet 通信ファンクションブロック

Ethernet 通信ファンクションブロックは，FCN/FCJ と Ethernet で接続された機器との通信機能を提供します。Ethernet 通信ファンクションブロックの一覧を以下に示します。本ファンクションブロックは，ライブラリ"SD_FCXPLCE_PF"で提供されます。

表 Ethernet 通信ファンクションブロック

ファンクションブロック	説明
SD_FCXPE_OPEN	通信タスクの起動
SD_FCXPE_SENDRECV	送受信要求

■ シリアル通信ファンクションブロック

シリアル通信ファンクションブロックは，FCN/FCJ と RS 通信で接続された機器との通信機能を提供します。シリアル通信ファンクションブロックの一覧を以下に示します。本ファンクションブロックは，ライブラリ"SD_FCXPLCS_PF"で提供されます。

表 シリアル通信ファンクションブロック

ファンクションブロック	説明
SD_FCXPM_OPEN	通信タスク起動処理
SD_FCXPM_OPEN_EX	拡張通信タスク起動処理
SD_FCXPM_SENDRECV	送受信処理
SD_FCXPM_SEND	送信処理
SD_FCXPM_RECV	受信処理
SD_FCXPM_GETCH	チャネル番号取得処理
SD_FCXPS_OPEN	通信タスク起動処理 (*1)
SD_FCXPS_SENDRECV	送受信処理 (*1)
SD_FCXPS_SEND	送信処理(*1)
SD_FCXPS_RECV	受信処理(*1)

*1: 旧版ファンクションブロック

■ システムユーティリティファンクションブロック

システムユーティリティファンクションブロックは，システム処理の補助機能として提供します。システムユーティリティファンクションブロックの一覧を以下に示します。本ファンクションブロックは，ライブラリ"SD_FCXSYS_LIB"で提供されます。

表 システムユーティリティファンクションブロック

ファンクションブロック	説明
SD_FCXS_ANY_TO_ANY	データ操作ファンクション

付録 D NPAS POUとPAS POUとの差分

NPAS POU の PAS POU からの機能の変更点と他 POU との組み合わせ時の注意について説明します。

■ NPAS POU の PAS POU からの機能の変更点

● データの持ち方の変更

NPAS POU と PAS POU は，どちらも制御データに CData_REAL 型のデータを使用していますが，Value メンバのデータの持ち方が違っています。
PAS POU では，Value メンバにスケール上下限値で正規化された 0～100%の値を使用していますが，NPAS POU では，Value メンバに工業単位に基づく工業量の値を使用しています。
NPAS POU では，工業量の値を使用することにより，エンジニアリング性を改善しています。

NPAS POU と PAS POU は，同じ CData_REAL 型のデータを使用しているため，プログラミング上では相互に接続することは可能ですが，データの内容が違っているため，そのままでは正常に動作しません。
また，別々の制御ループで，NPAS POU と PAS POU を同時に使用することができますが，シーケンスなどからのデータを設定する場合に設定する値が異なり，非常に煩雑になります。

重 要

同一プロジェクト内では，NPAS POU と PAS POU は混在して使用しないでください。

● アクセスパラメータの参照・設定方法の変更

PAS POU では，アクセスパラメータの参照と設定に各々別の変数を割り当て，参照や設定を行う必要がありました。また，アクセスパラメータの設定では，アクセスパラメータ設定用の特殊なファンクションを使用する必要がありました。
NPAS POU では，アクセスパラメータの参照と設定を同一の変数で行え，MOVE ファンクションでのアクセスパラメータの設定が可能になりました。

● エンジニアリングパラメータの設定方法の変更

PAS POU では，エンジニアリングパラメータの設定に，エンジニアリングパラメータ設定用のファンクションブロックを PAS POU インスタンス（タグ）ごとに用意する必要がありました。
NPAS POU では，アクセスパラメータの設定と同様にエンジニアリングパラメータ設定用の変数を割り当て，その変数のメンバへ設定値を代入することで，エンジニアリングパラメータの設定を行えるようになりました。

● ウォッチウィンドウによるデータ変更可能

PAS POU では，デバッグ時にウォッチウィンドウから，アクセスパラメータの値の変更ができませんでした。
NPAS POU では，アクセスパラメータ参照用変数と設定用変数を同一化したことにより，ウォッチウィンドウから，アクセスパラメータの値の変更が可能になりました。
この結果，NPAS POU では，ロジックデザイナ上での動作のデバッグが可能になっています。

■ 他 POU との組み合わせ時の注意

以下のアプリケーションポートフォリオは，データに正規化データを使用しています。このアプリケーションポートフォリオの POU と組み合わせて使用する場合は，PAS POU と組み合わせて使用するか，データを正規化データから工業量データに変換して，NPAS POU に接続してください。

・電力モニタ通信ポートフォリオ
・温調計通信ポートフォリオ

Technical Information 改訂情報

資料名称　　：NPAS POU 説明書（概要編）
資料番号　　：TI 34P02P25-01

2005 年 5 月／初版／R1.60 以降
新規発行
- IM 34P02P25-01 「NPAS POU 説明書（概要編）2 版」を TI 化

2005 年 8 月／2 版／R1.70 以降
改訂
- オンラインダウンロードの説明追加
- 誤記訂正

2006 年 2 月／3 版／R1.80 以降
改訂
- 高分解能パルス幅出力処理 POU の追加

2008 年 7 月／4 版／R2.01 以降
改訂
- 入出力データ処理 POU ライセンスフリーの説明追加

2010 年 6 月／5 版／R3.01 以降
改訂
- IM 34P02P25-01 「NPAS POU 説明書（概要編）6 版」を TI 化

2011 年 3 月／6 版／R3.10 以降
改訂
- IM 34P02P25-01 「NPAS POU 説明書（概要編）7 版」を TI 化

2012 年 6 月／7 版／R3.20 以降
改訂
- IM 34P02P25-01 「NPAS POU 説明書（概要編）8 版」を TI 化

2014 年 3 月／8 版／R3.40 以降
改訂
- IM 34P02P25-01 「NPAS POU 説明書（概要編）9 版」を TI 化（NPAS POU の追加）

■お問い合わせについて
本書の内容に関するご質問は，下記メールアドレスにて
お願いいたします。
問い合わせメールアドレス：stardom_info@cs.jp.yokogawa.com
■著作者　横河電機株式会社
■発行者　横河電機株式会社
〒180-8750　東京都武蔵野市中町 2-9-32

Blank Page